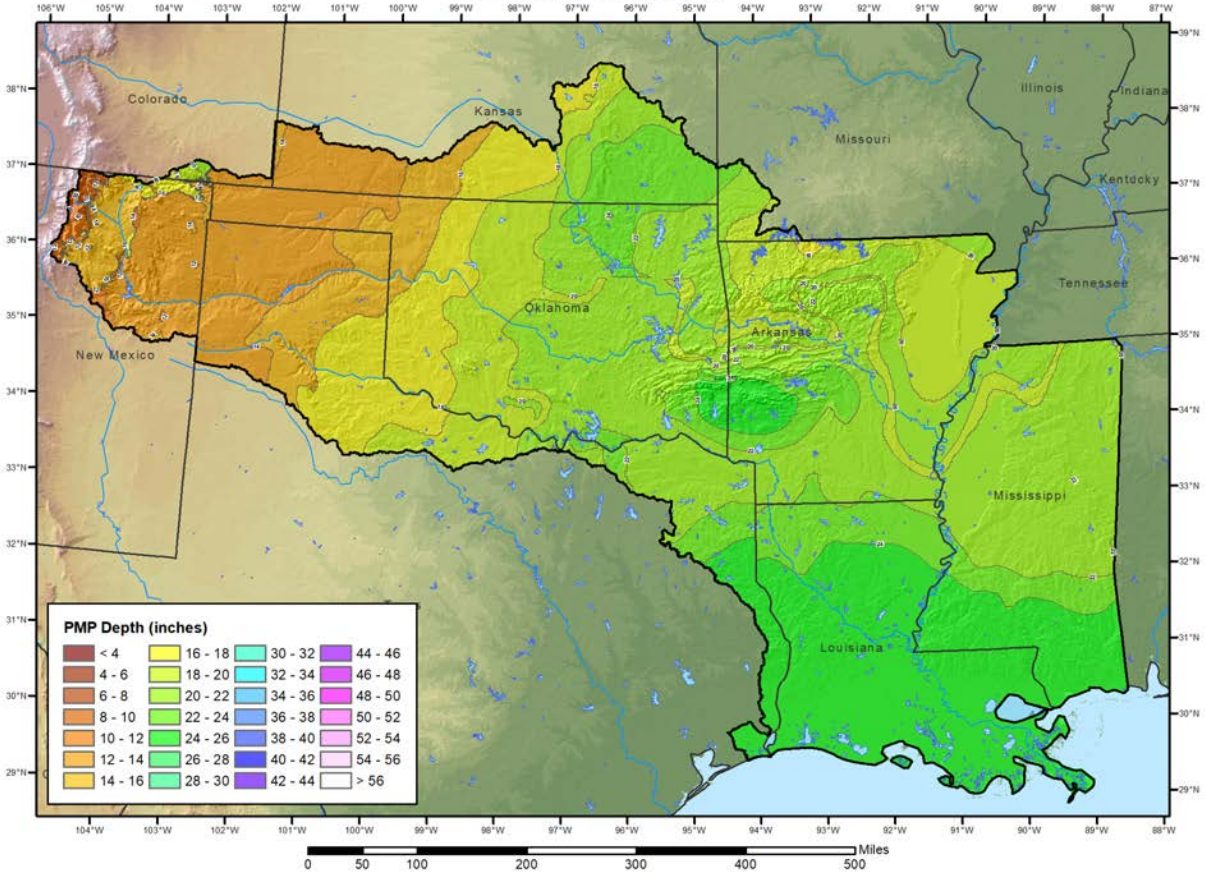
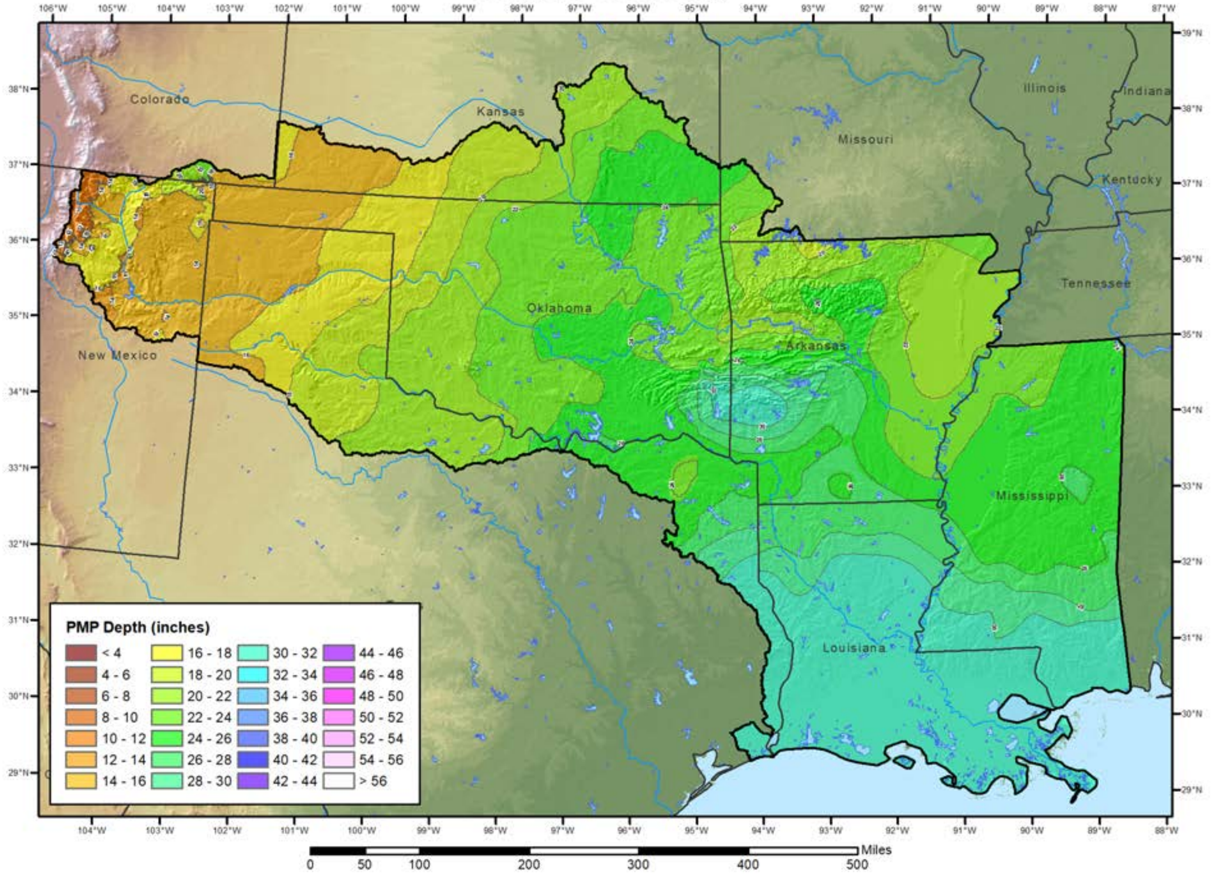
# Appendix A
## Probable Maximum Precipitation (PMP) Maps

# General Storms

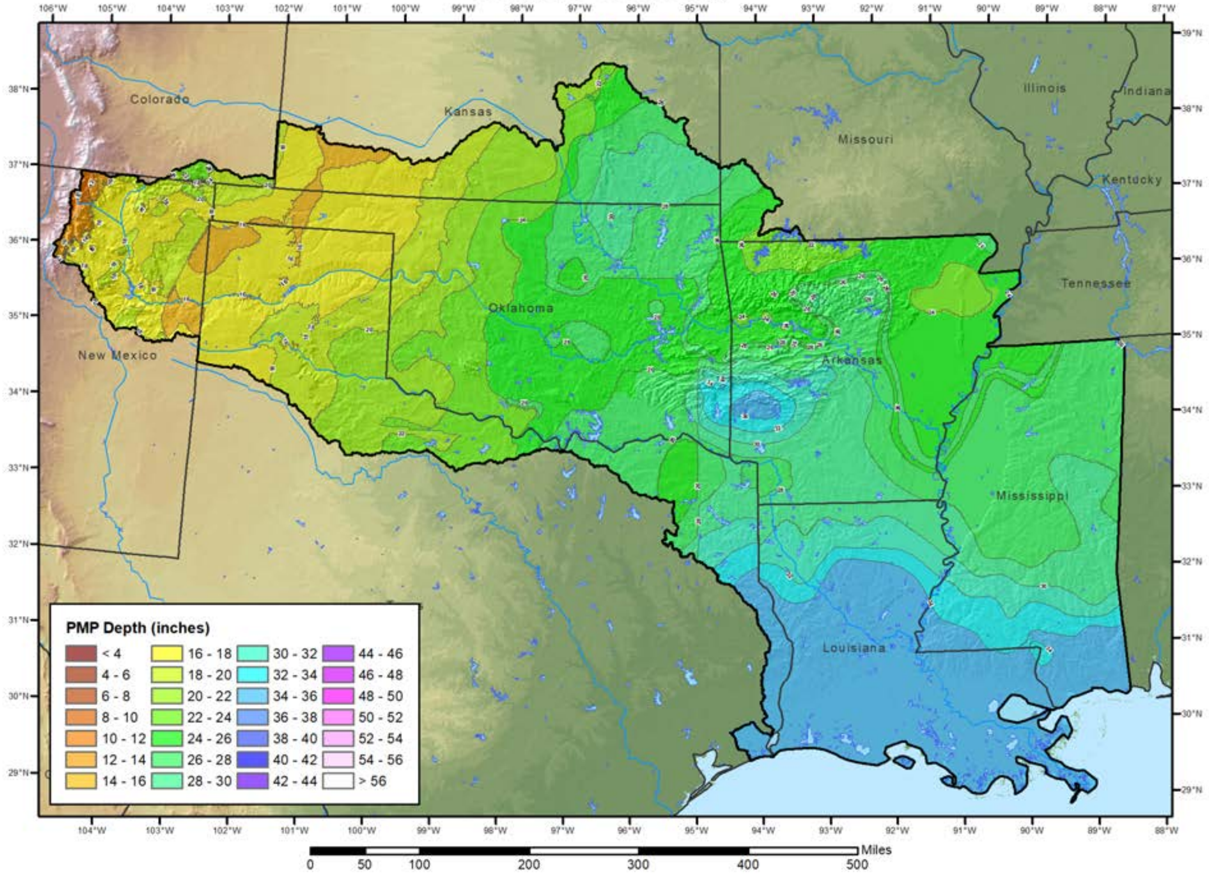
10 mi$^2$ 6-hour - General Storm PMP (in.)
AR-LA-MS-OK PMP Study



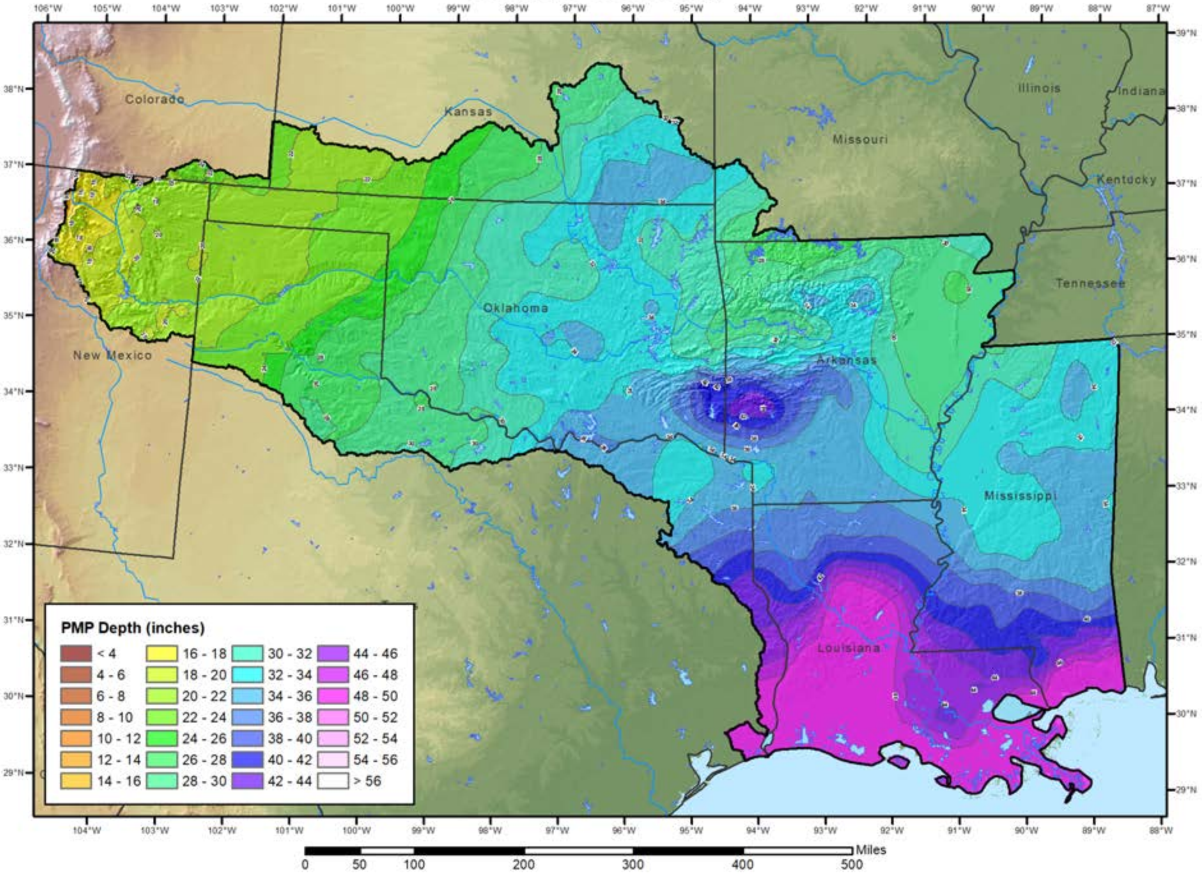
10 mi$^2$ 12-hour - General Storm PMP (in.)
AR-LA-MS-OK PMP Study



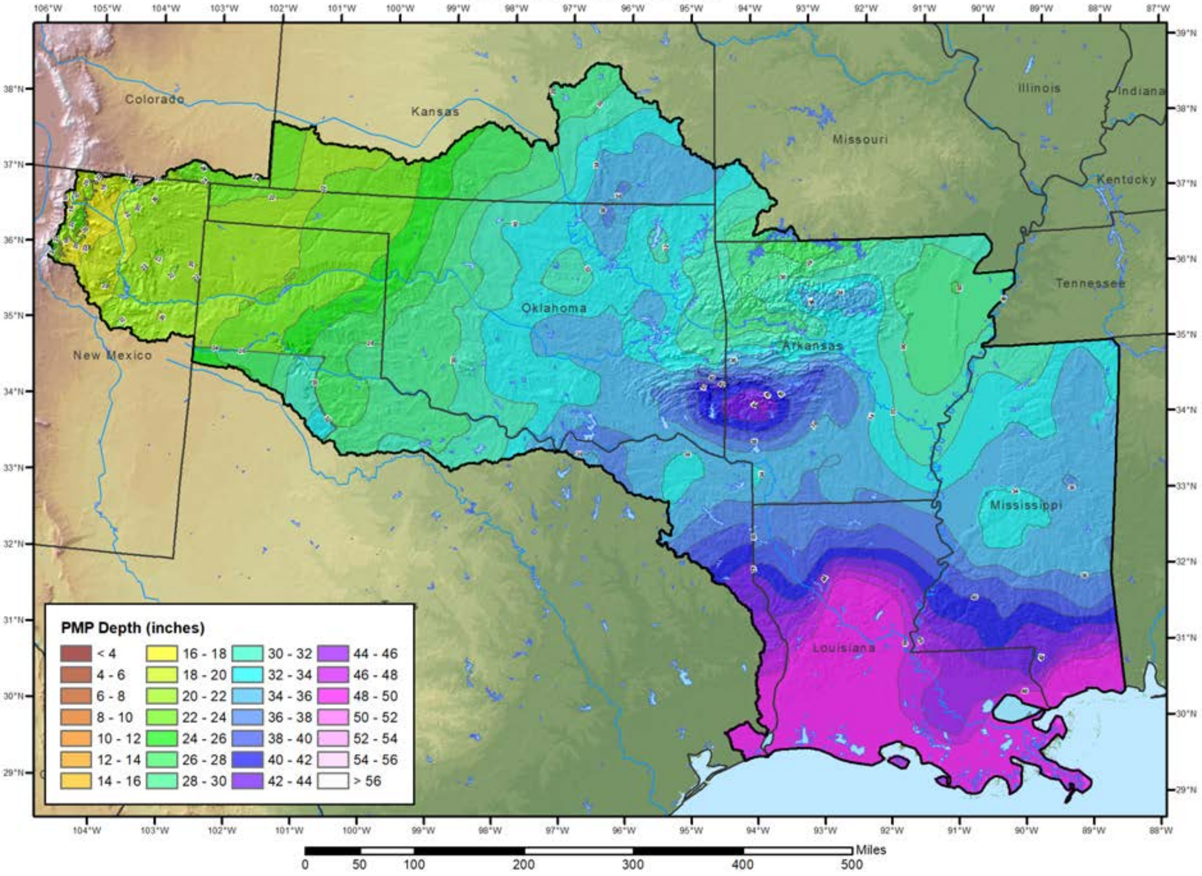
10 mi$^2$ 24-hour - General Storm PMP (in.)
AR-LA-MS-OK PMP Study



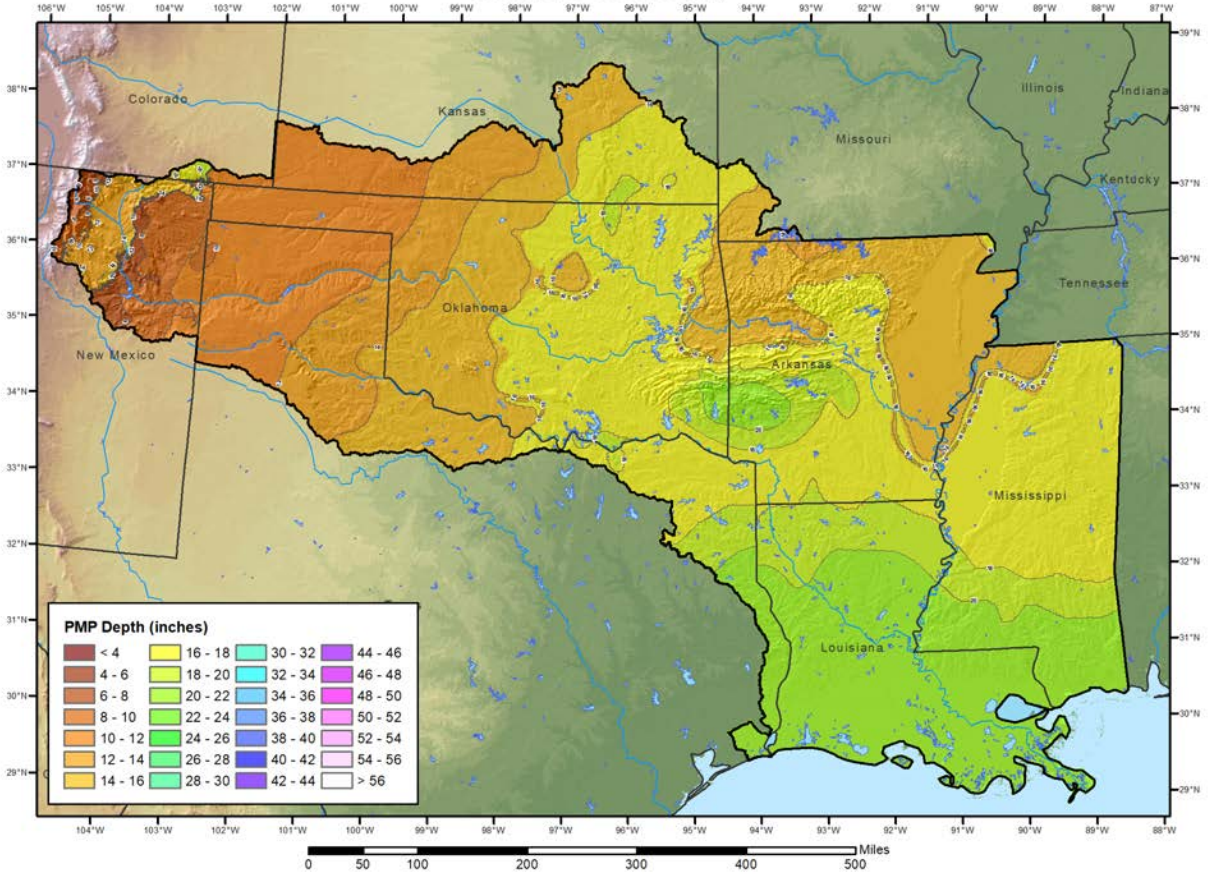
10 mi$^2$ 48-hour - General Storm PMP (in.)
AR-LA-MS-OK PMP Study



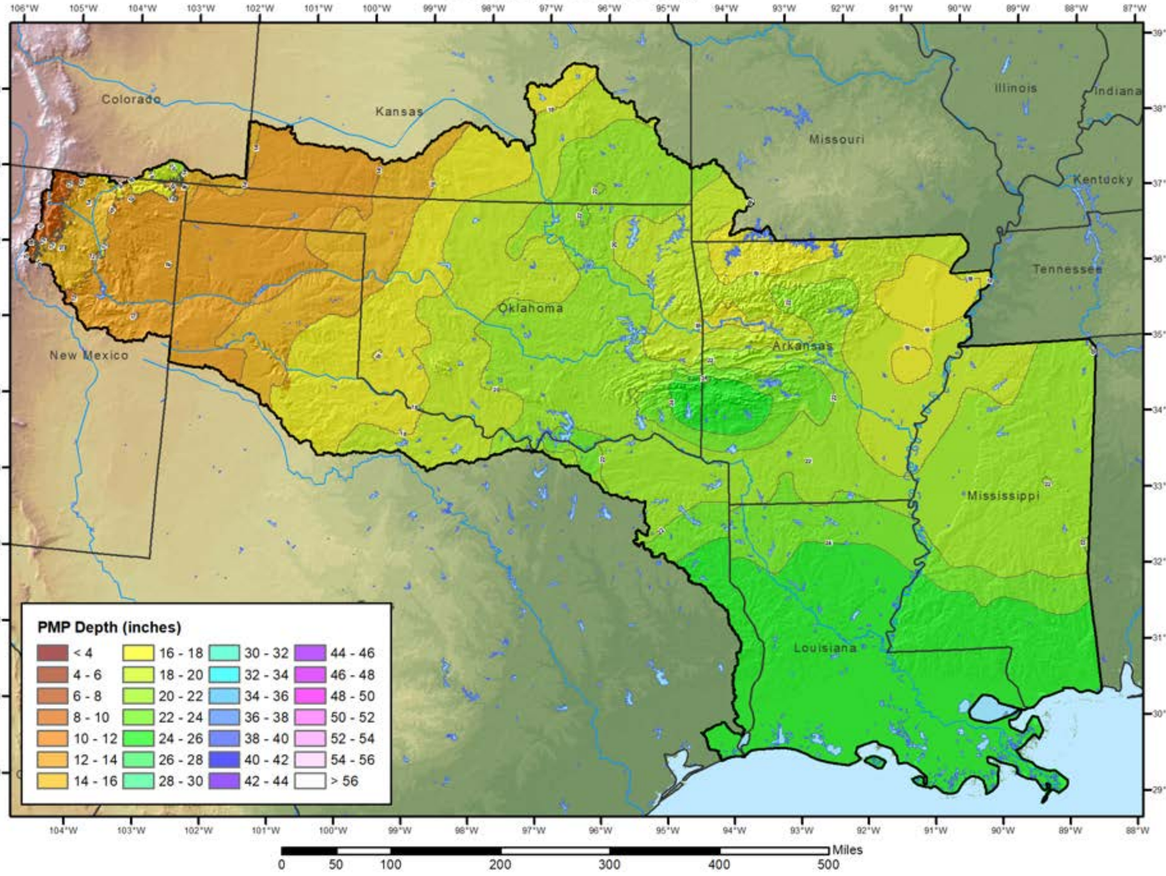
10 mi$^2$ 72-hour - General Storm PMP (in.)
AR-LA-MS-OK PMP Study



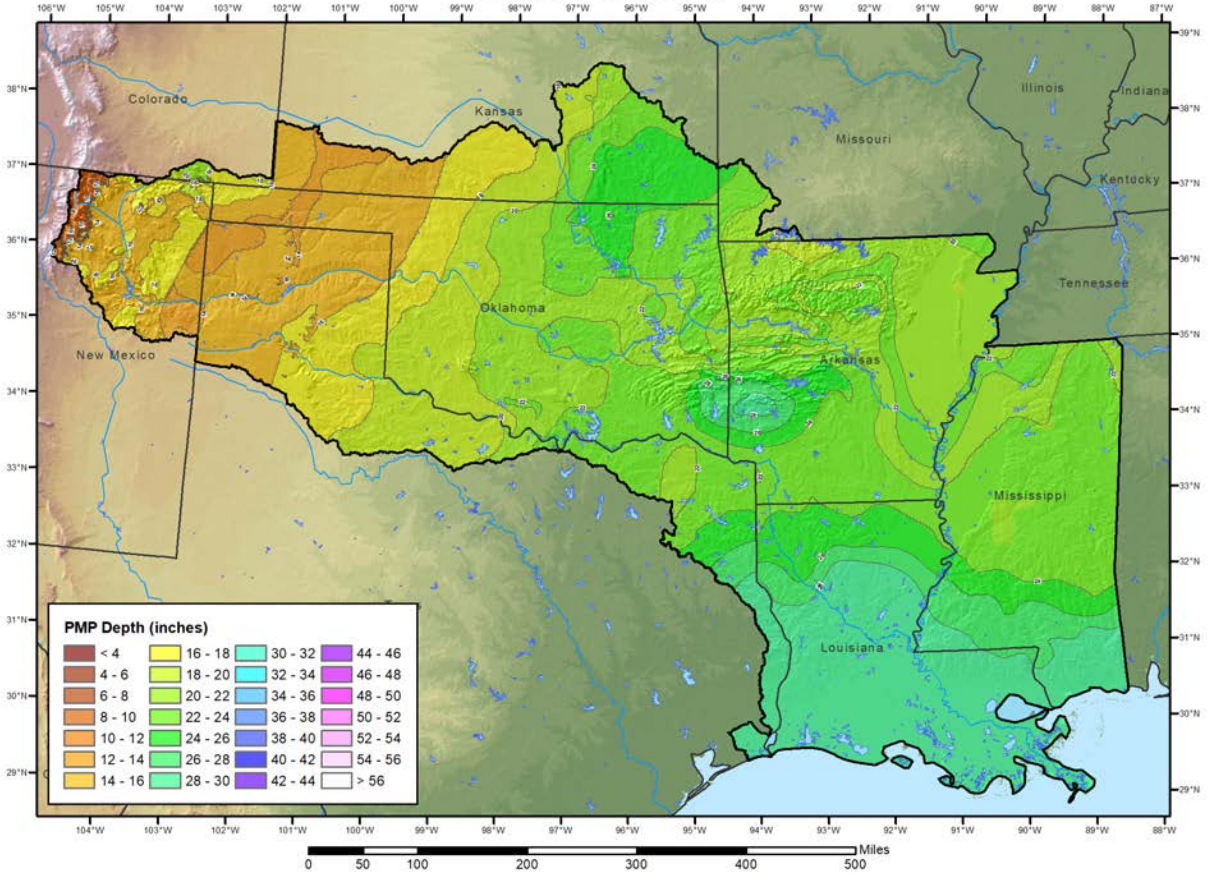
200 mi$^2$ 6-hour - General Storm PMP (in.)
AR-LA-MS-OK PMP Study



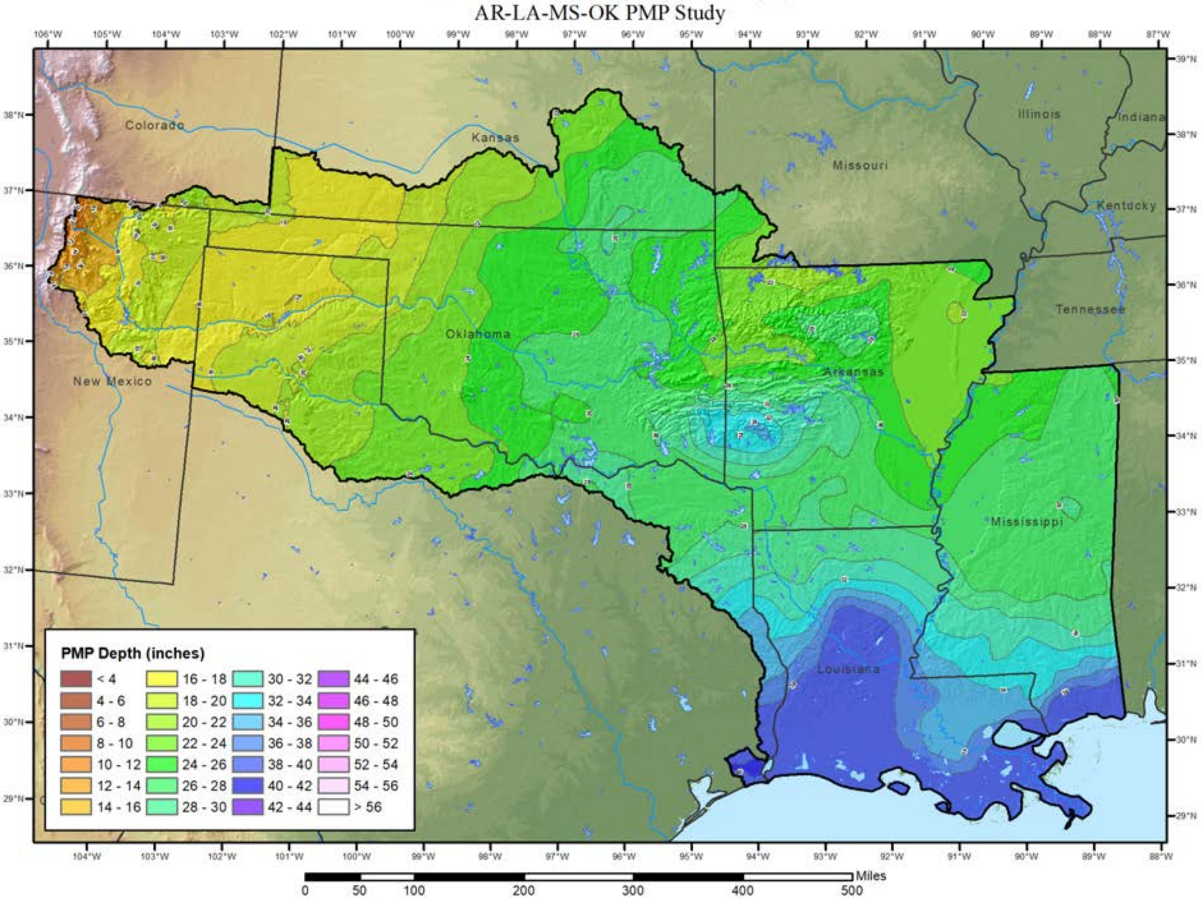
200 mi$^2$ 12-hour - General Storm PMP (in.)
AR-LA-MS-OK PMP Study



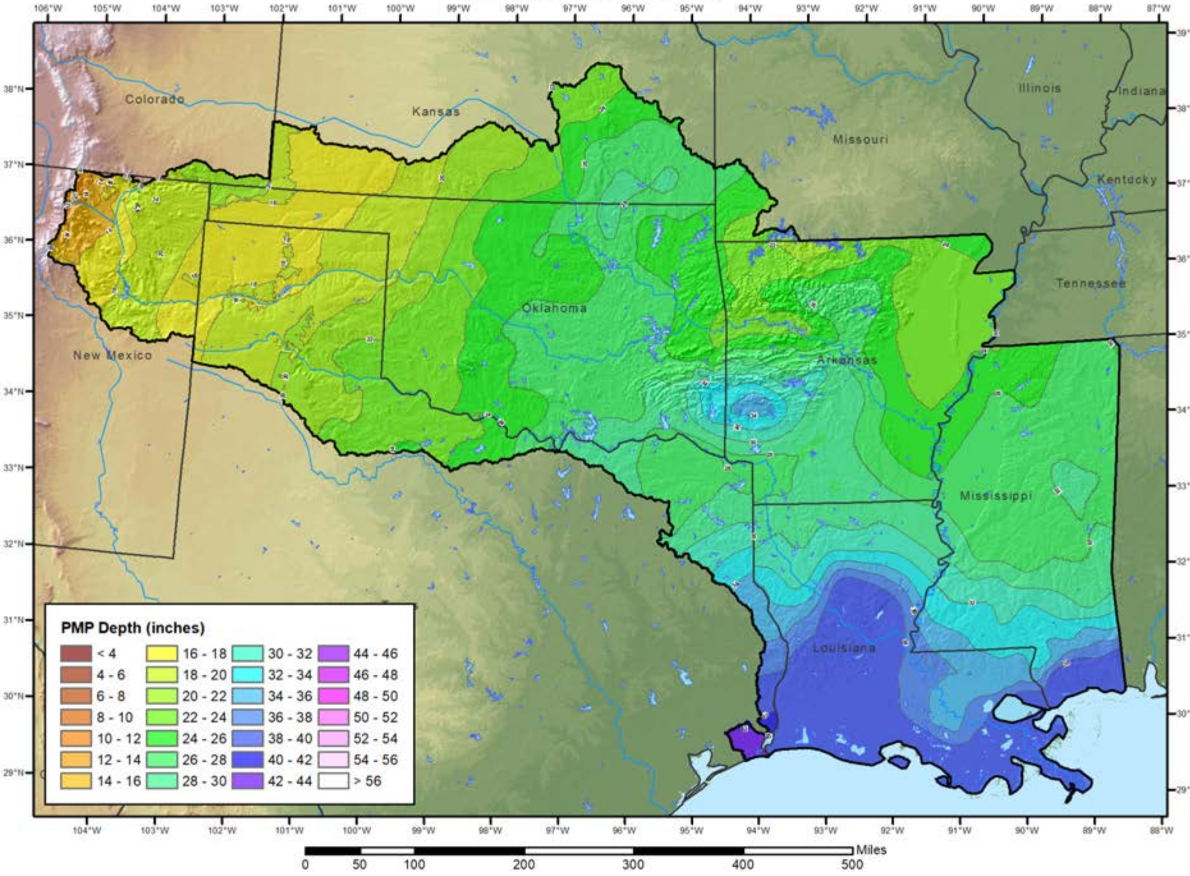
200 mi$^2$ 24-hour - General Storm PMP (in.)
AR-LA-MS-OK PMP Study



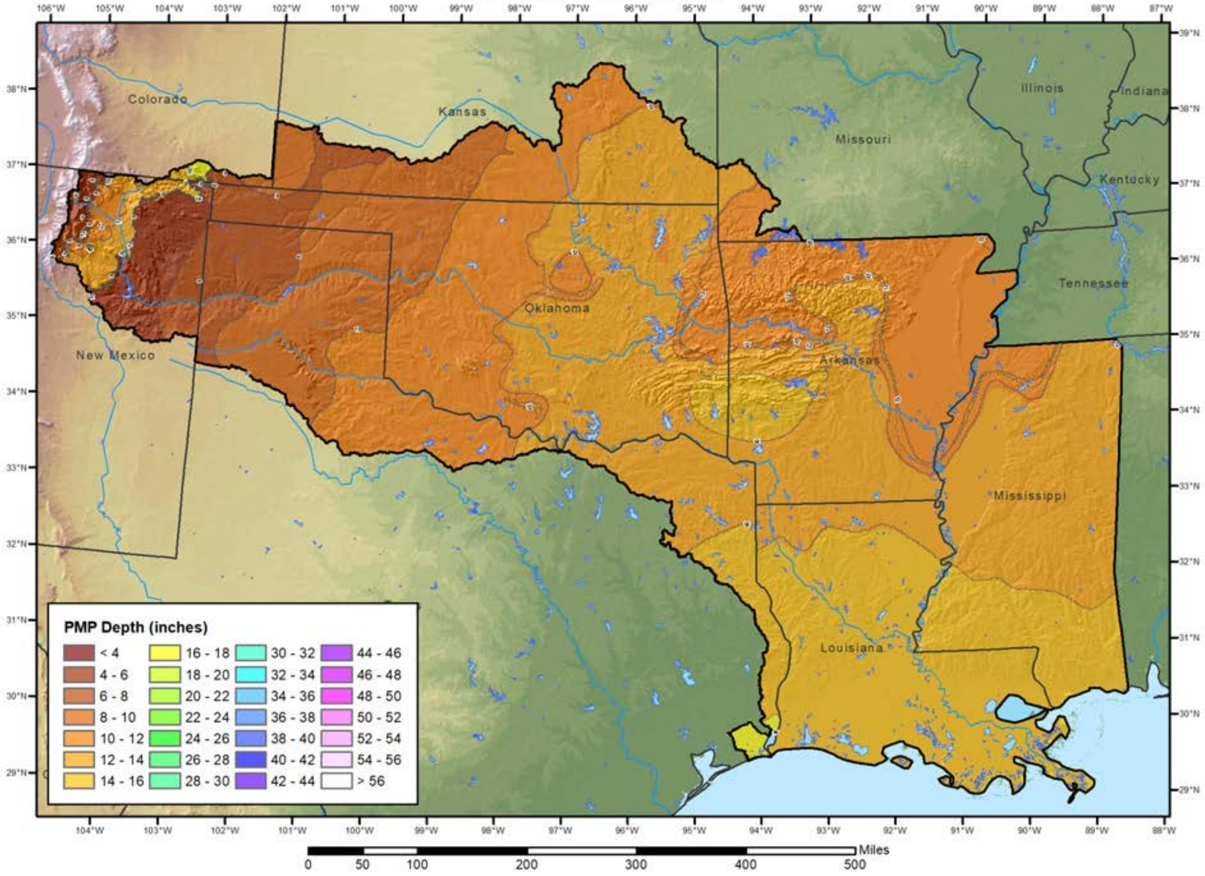
200 mi$^2$ 48-hour - General Storm PMP (in.)
AR-LA-MS-OK PMP Study



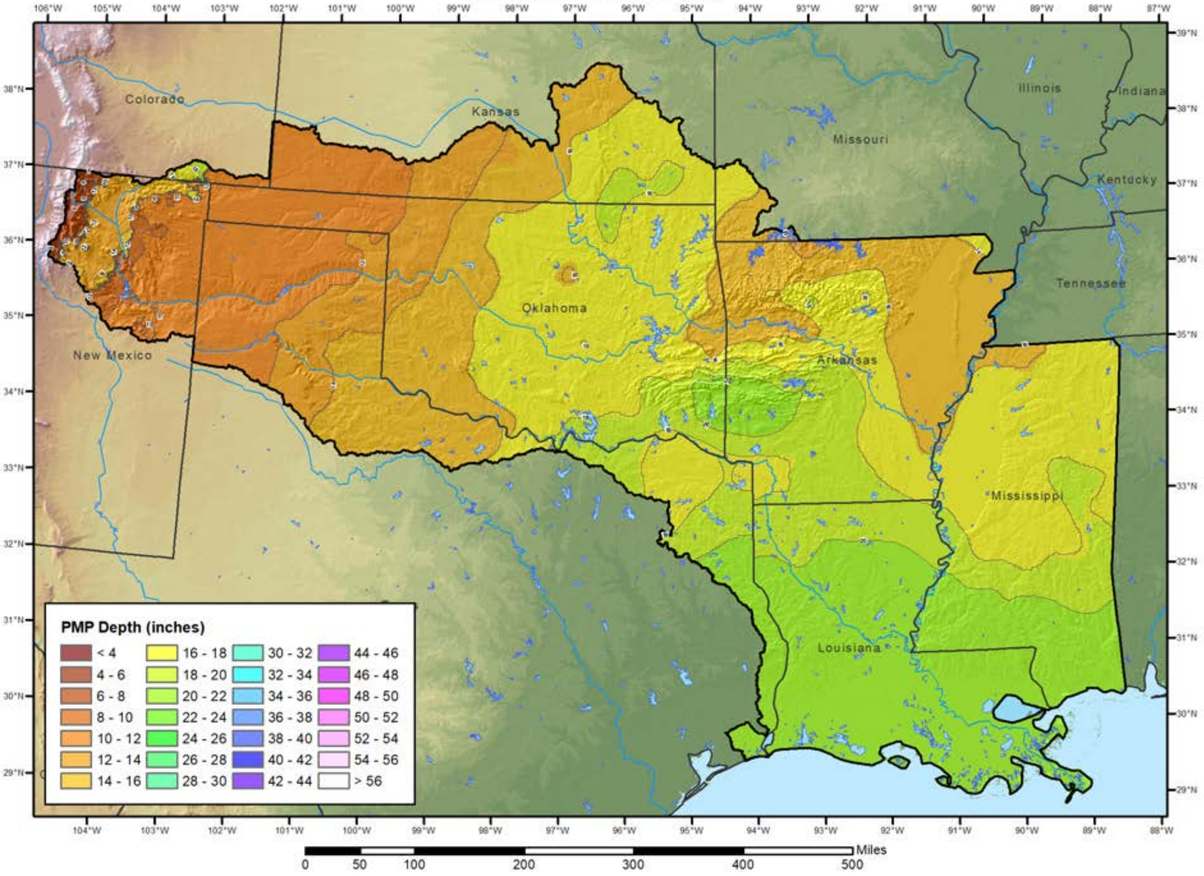
200 mi$^2$ 72-hour - General Storm PMP (in.)
AR-LA-MS-OK PMP Study



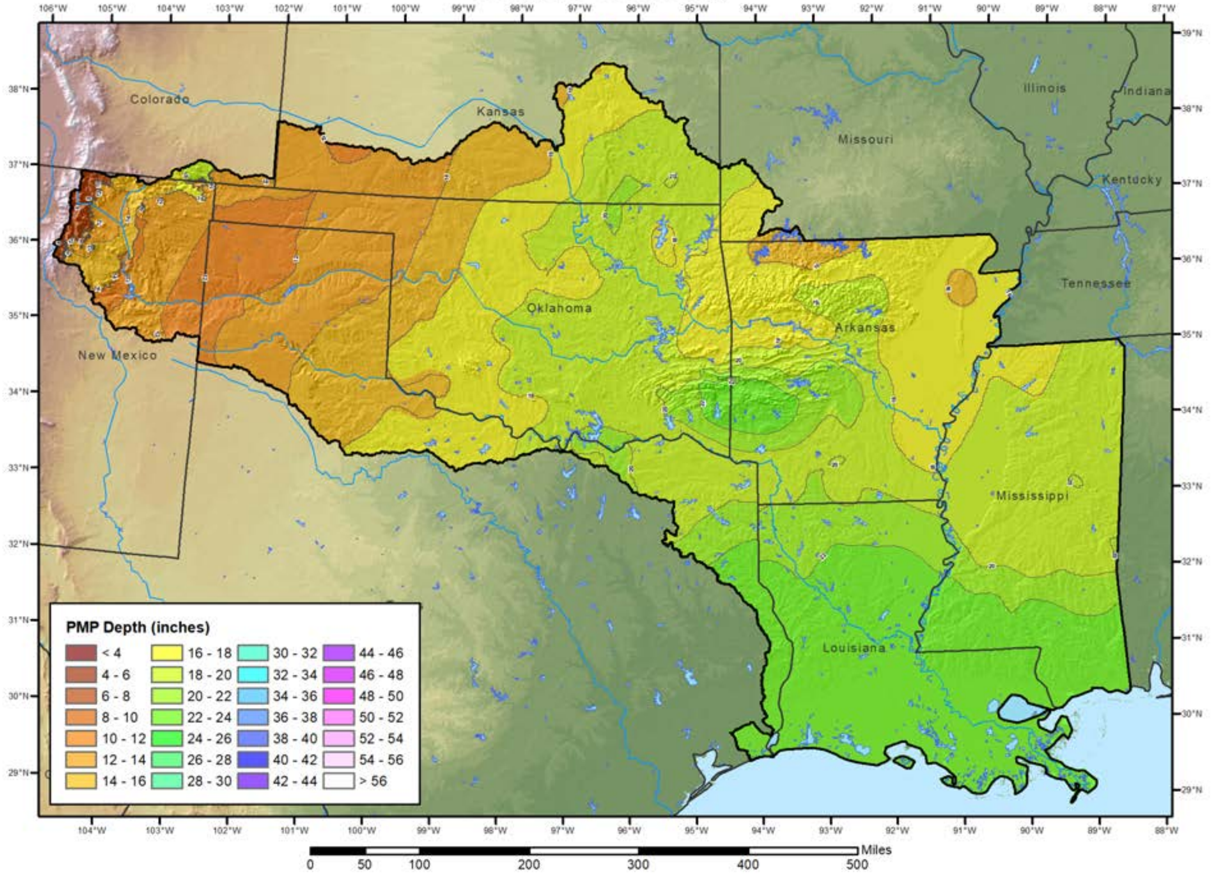
1,000 mi$^2$ 6-hour - General Storm PMP (in.)
AR-LA-MS-OK PMP Study



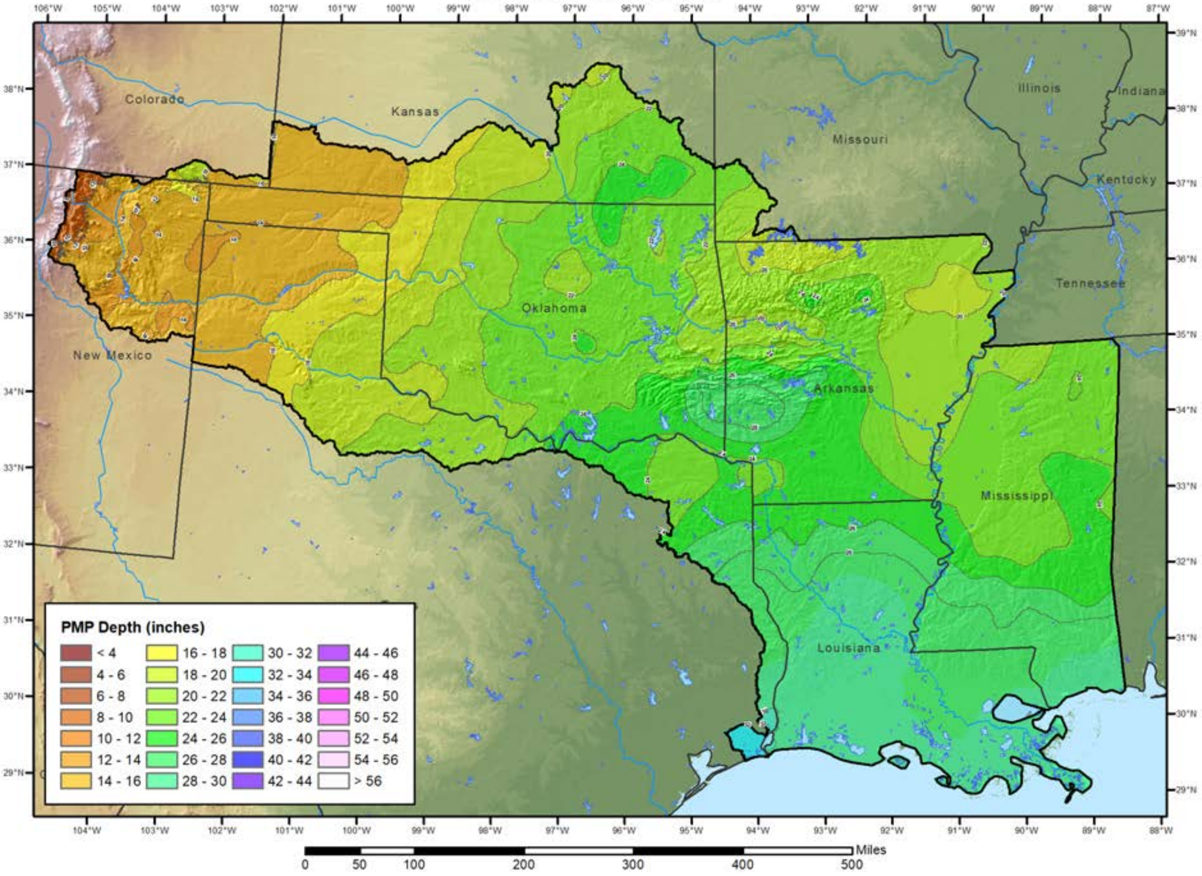
1,000 mi$^2$ 12-hour - General Storm PMP (in.)
AR-LA-MS-OK PMP Study



1,000 mi$^2$ 24-hour - General Storm PMP (in.)
AR-LA-MS-OK PMP Study

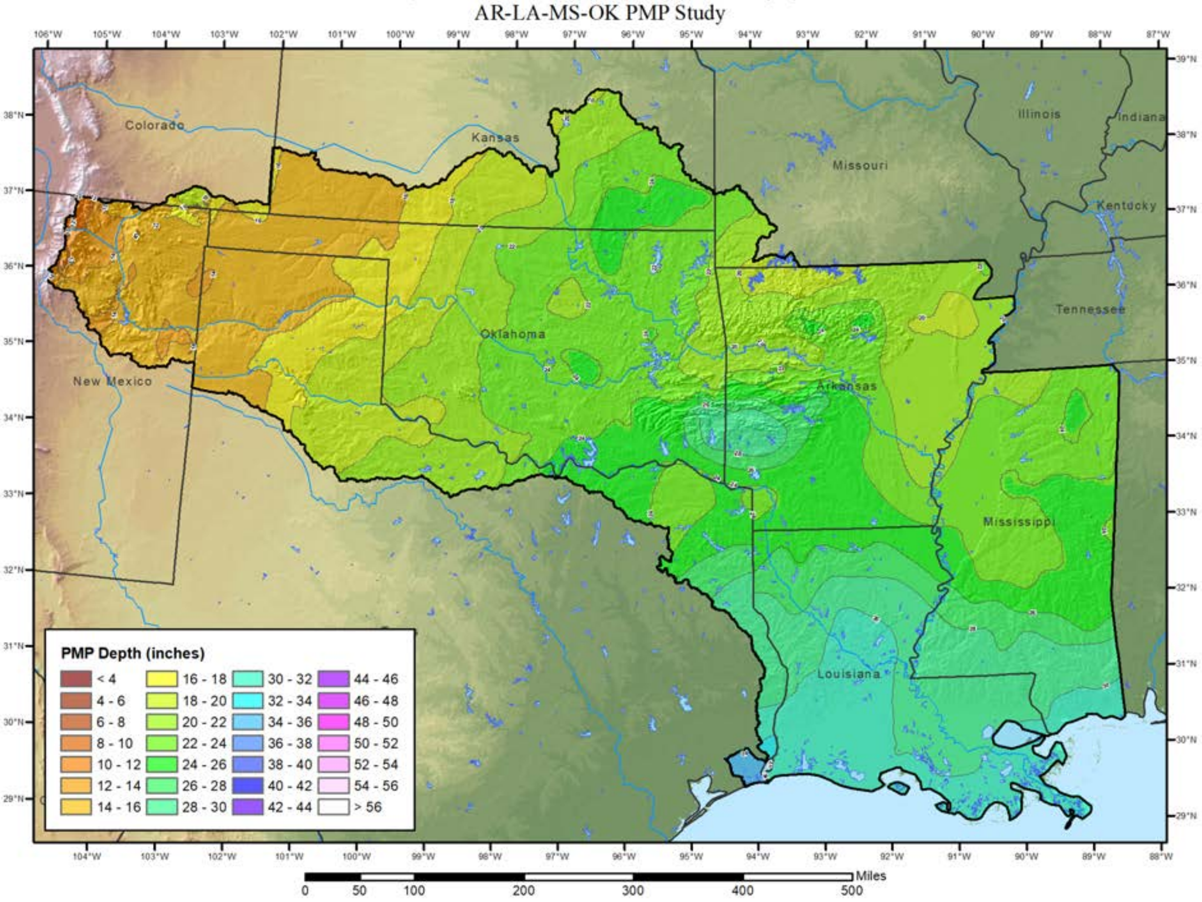


1,000 mi$^2$ 48-hour - General Storm PMP (in.)
AR-LA-MS-OK PMP Study

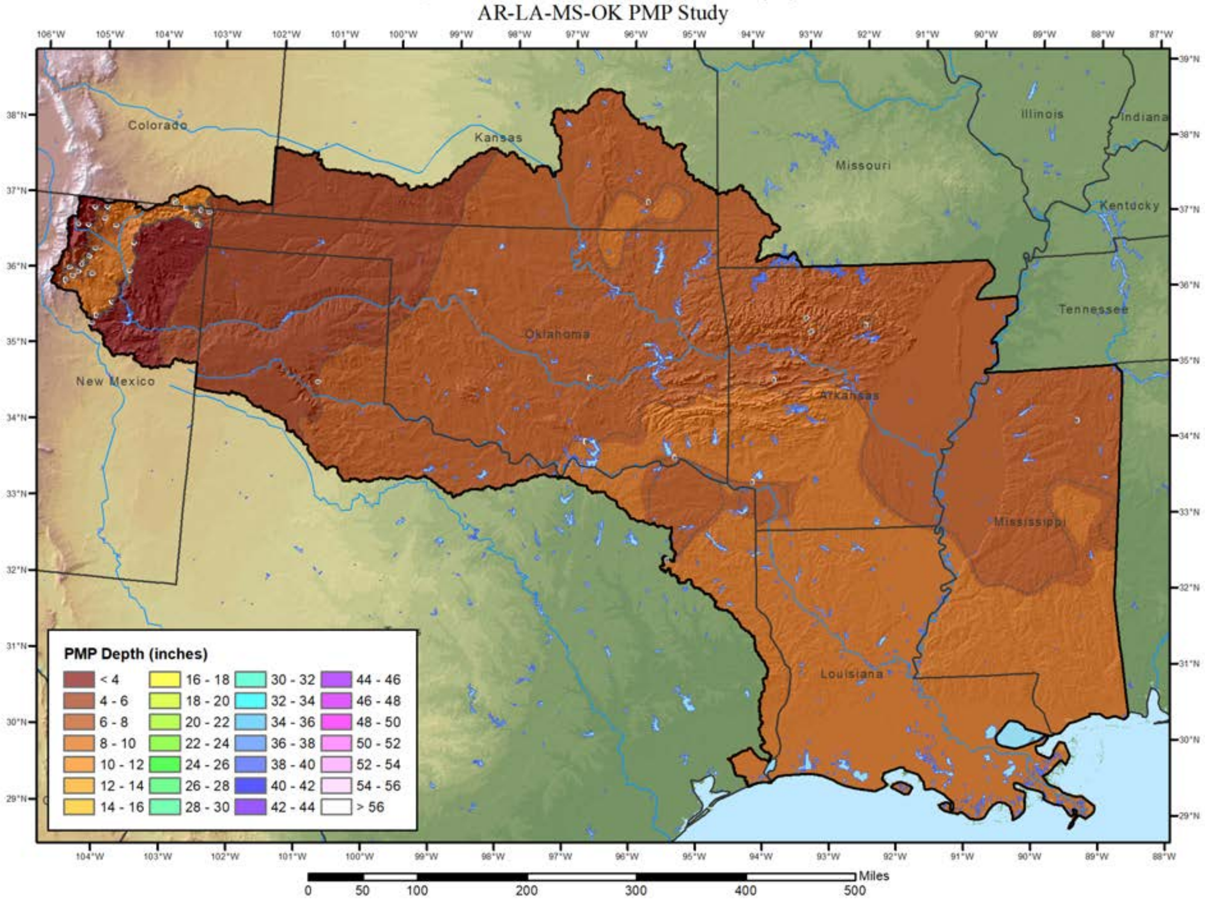


1,000 mi$^2$ 72-hour - General Storm PMP (in.)

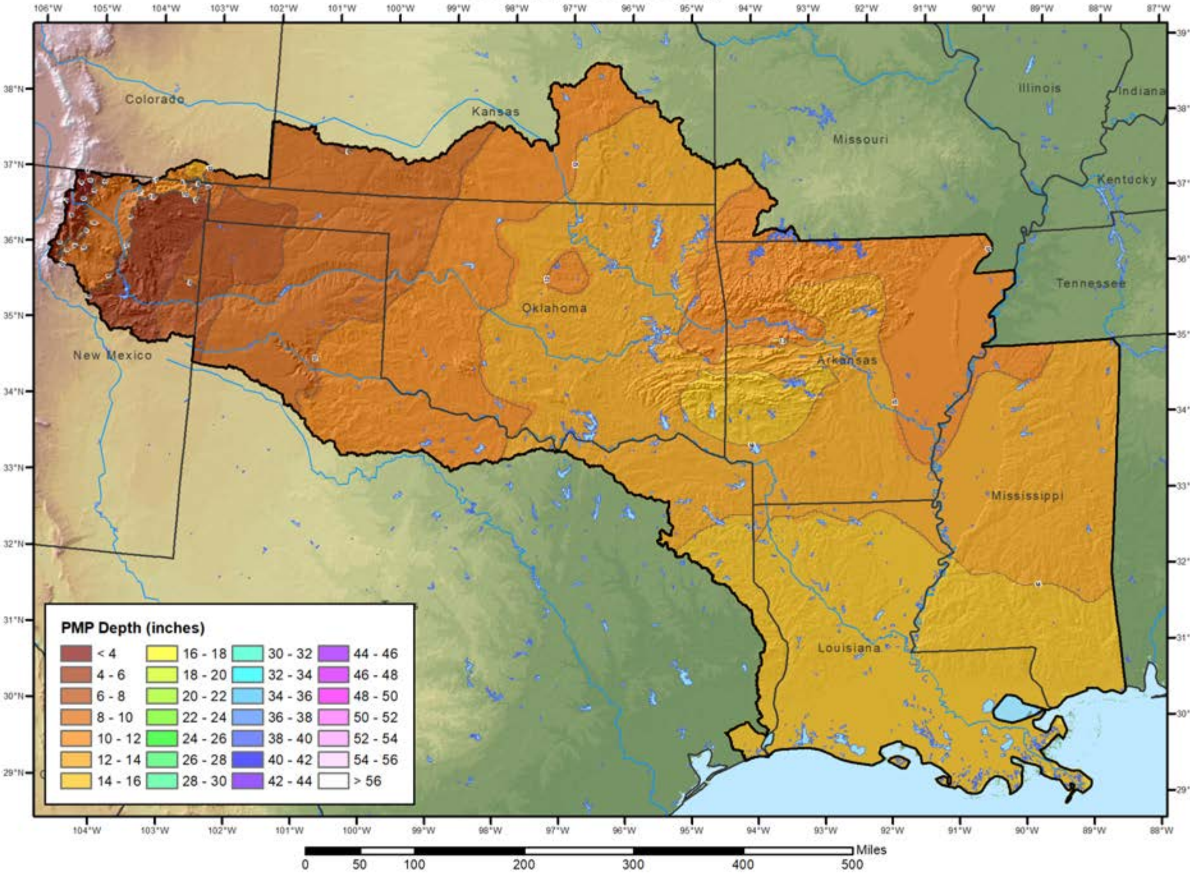
AR-LA-MS-OK PMP Study



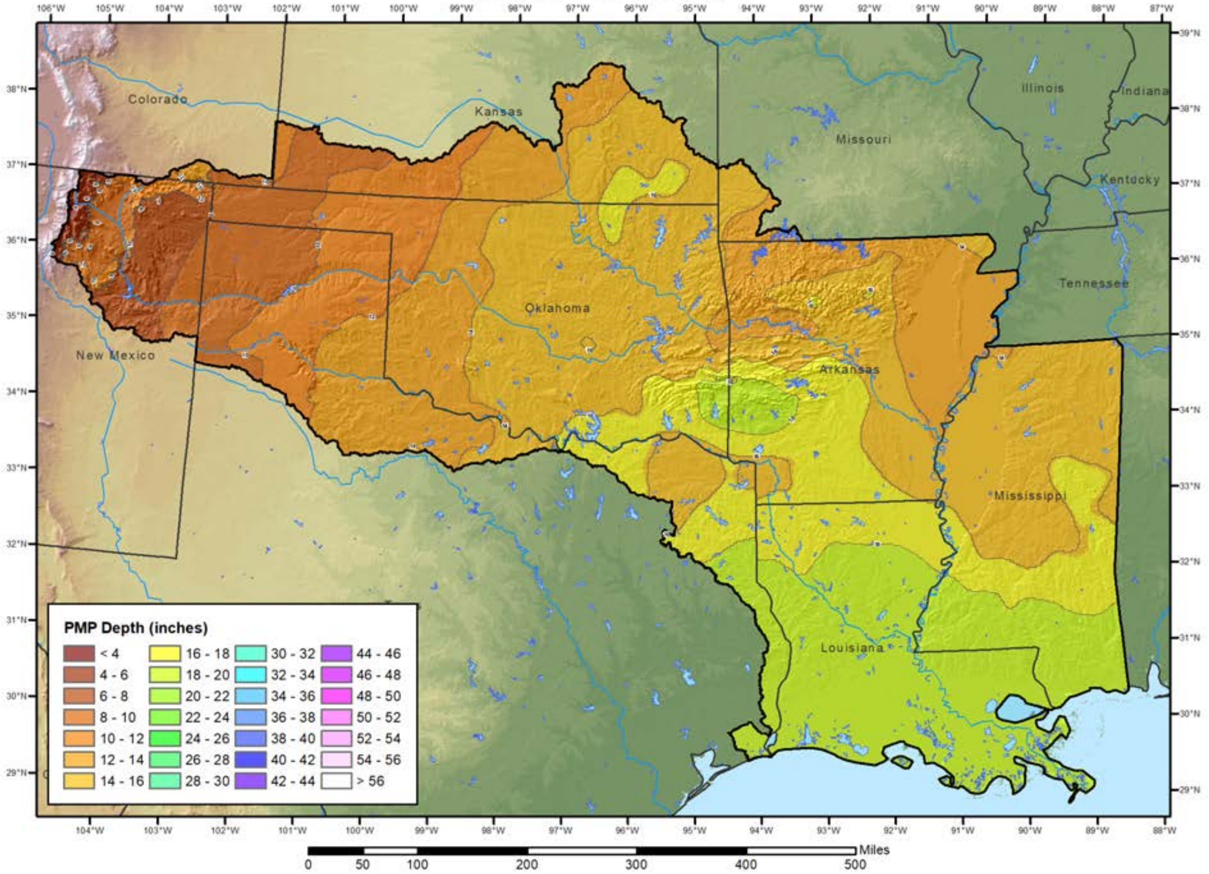
5,000 mi$^2$ 6-hour - General Storm PMP (in.)
AR-LA-MS-OK PMP Study



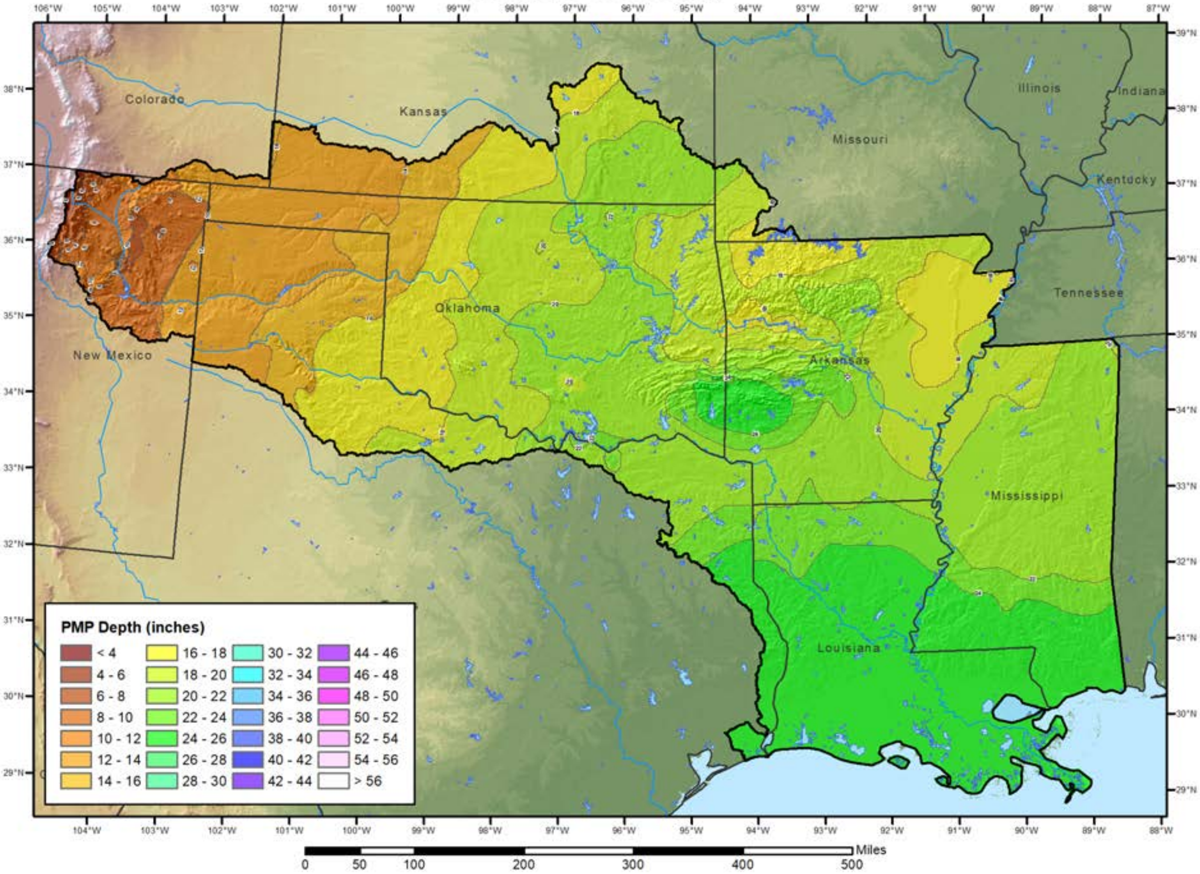
5,000 mi$^2$ 12-hour - General Storm PMP (in.)
AR-LA-MS-OK PMP Study



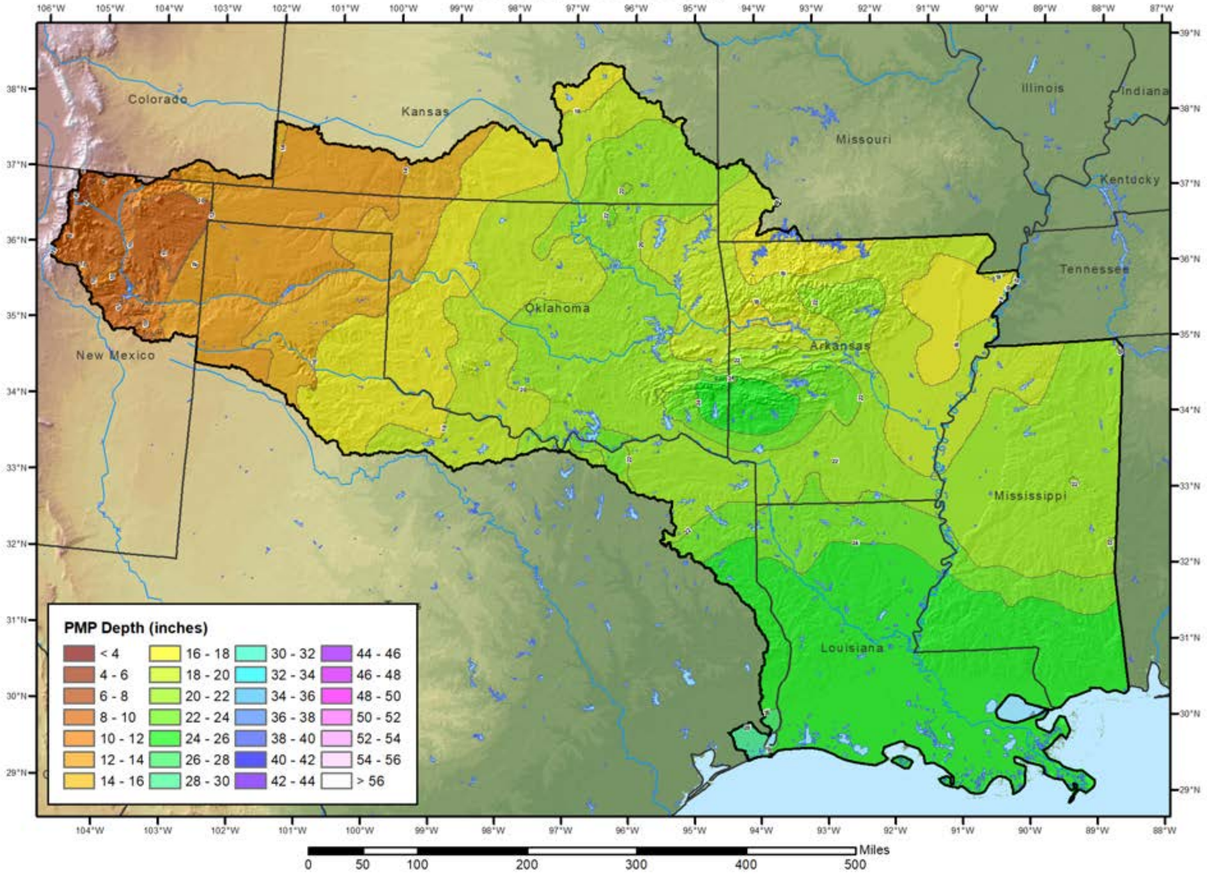
5,000 mi$^2$ 24-hour - General Storm PMP (in.)
AR-LA-MS-OK PMP Study



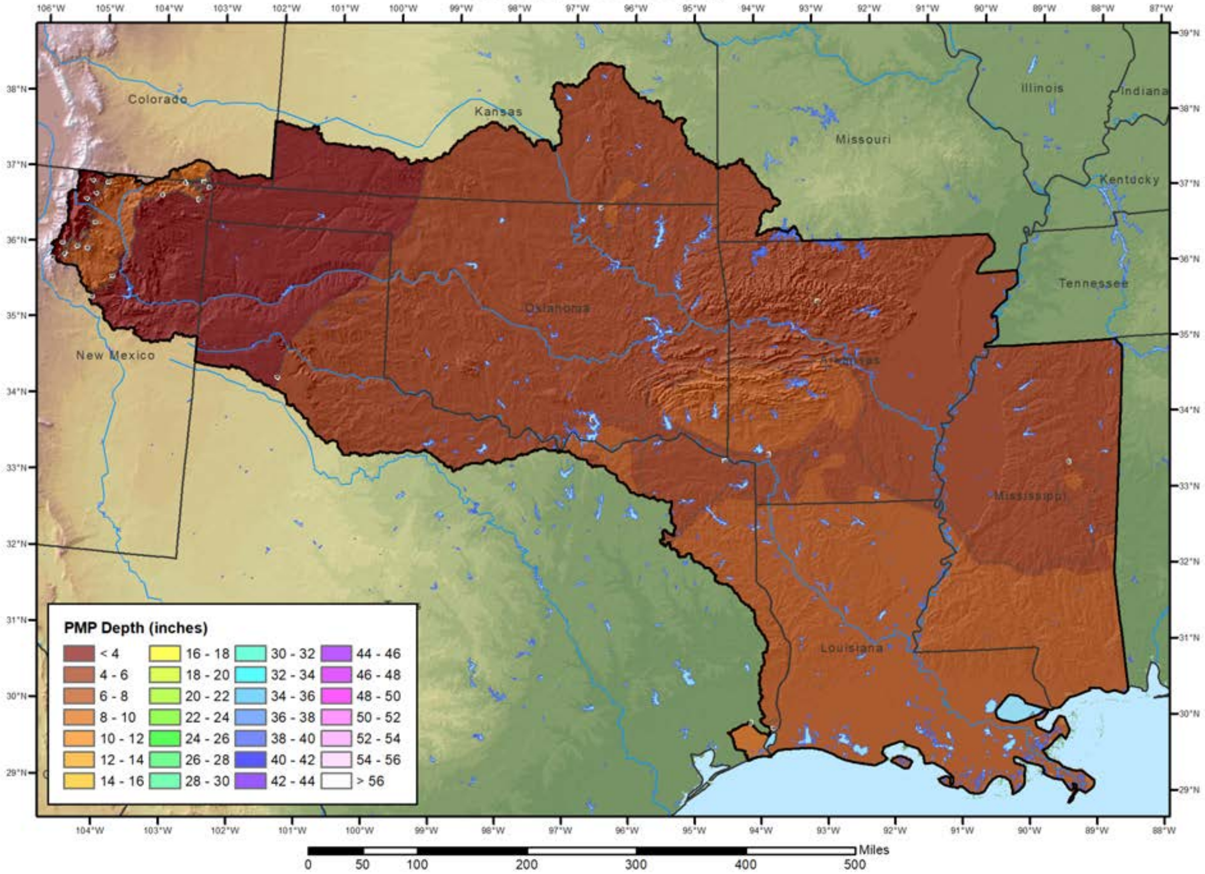
5,000 mi$^2$ 48-hour - General Storm PMP (in.)
AR-LA-MS-OK PMP Study



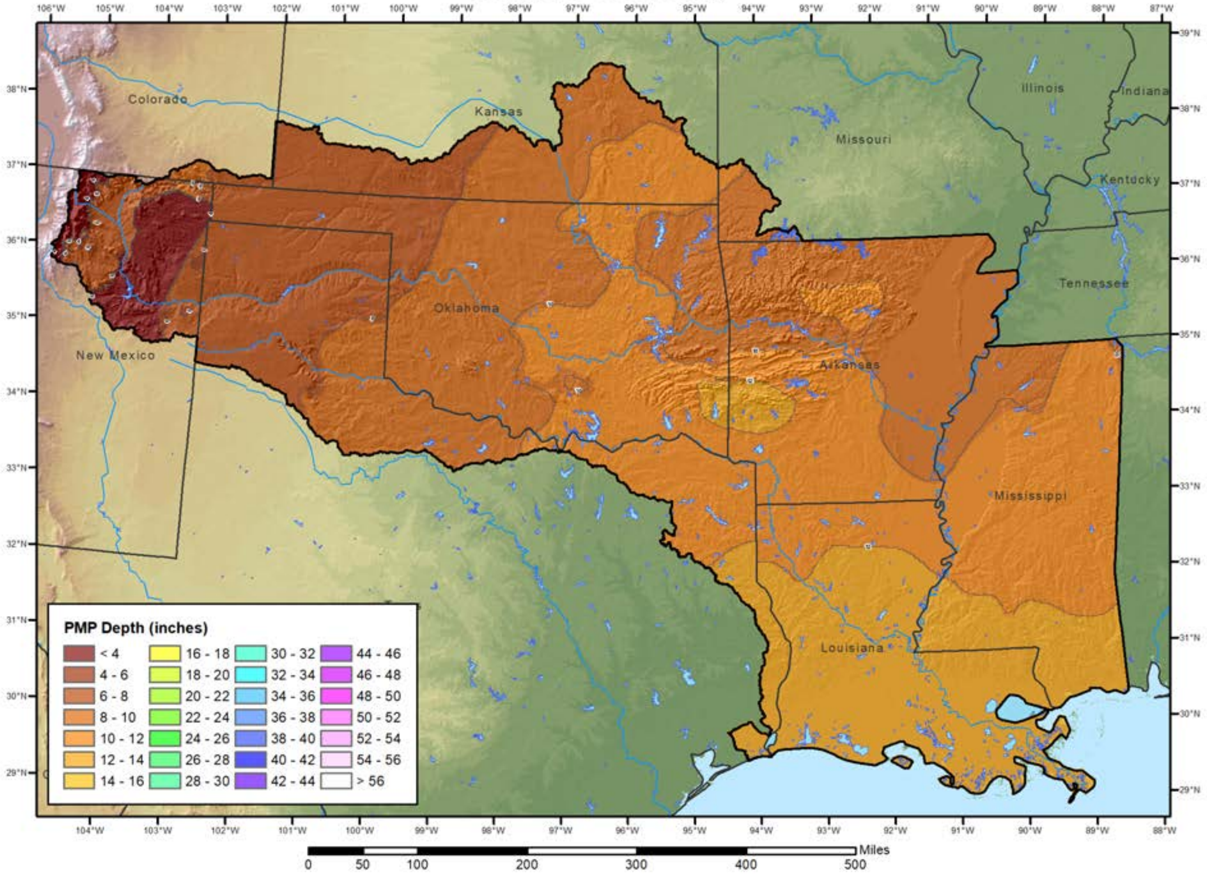
5,000 mi$^2$ 72-hour - General Storm PMP (in.)
AR-LA-MS-OK PMP Study



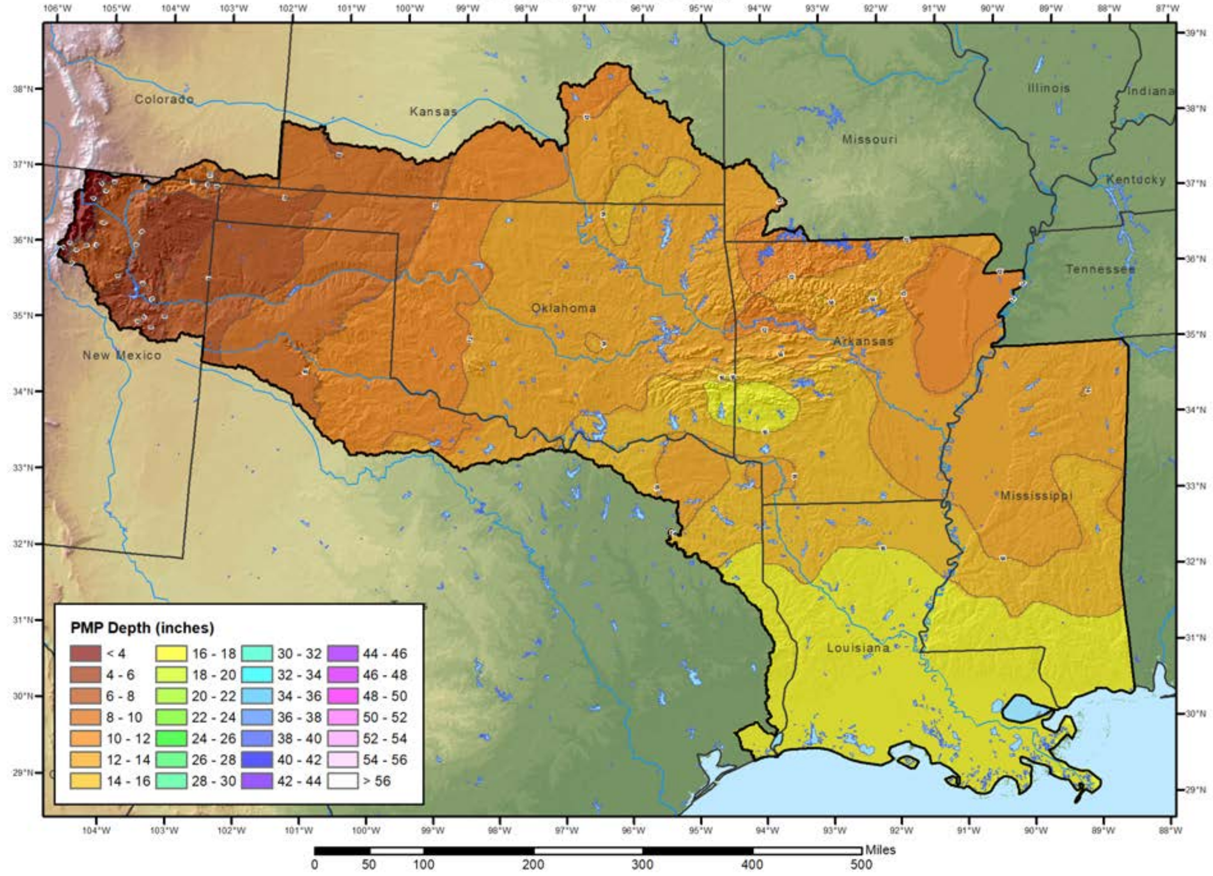
10,000 mi$^2$ 6-hour - General Storm PMP (in.)
AR-LA-MS-OK PMP Study



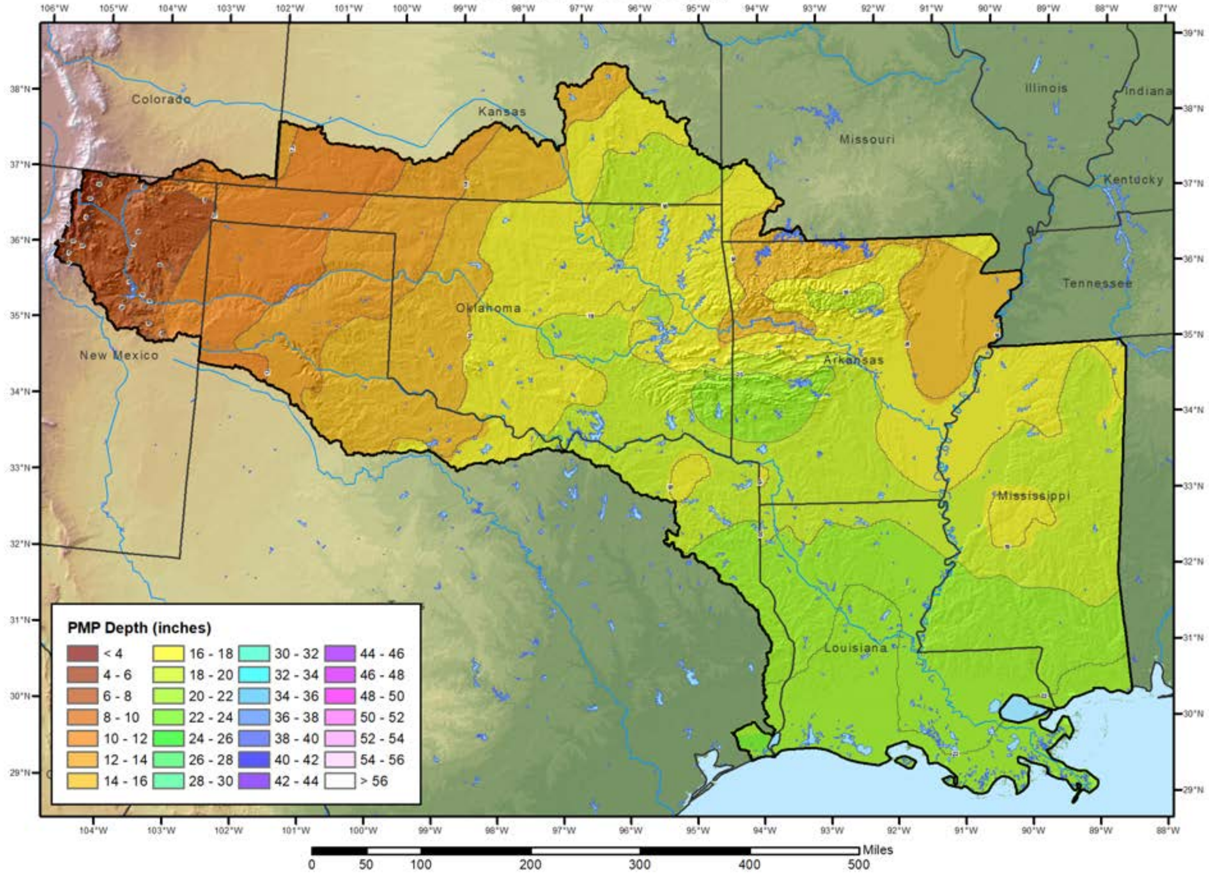
10,000 mi$^2$ 12-hour - General Storm PMP (in.)
AR-LA-MS-OK PMP Study



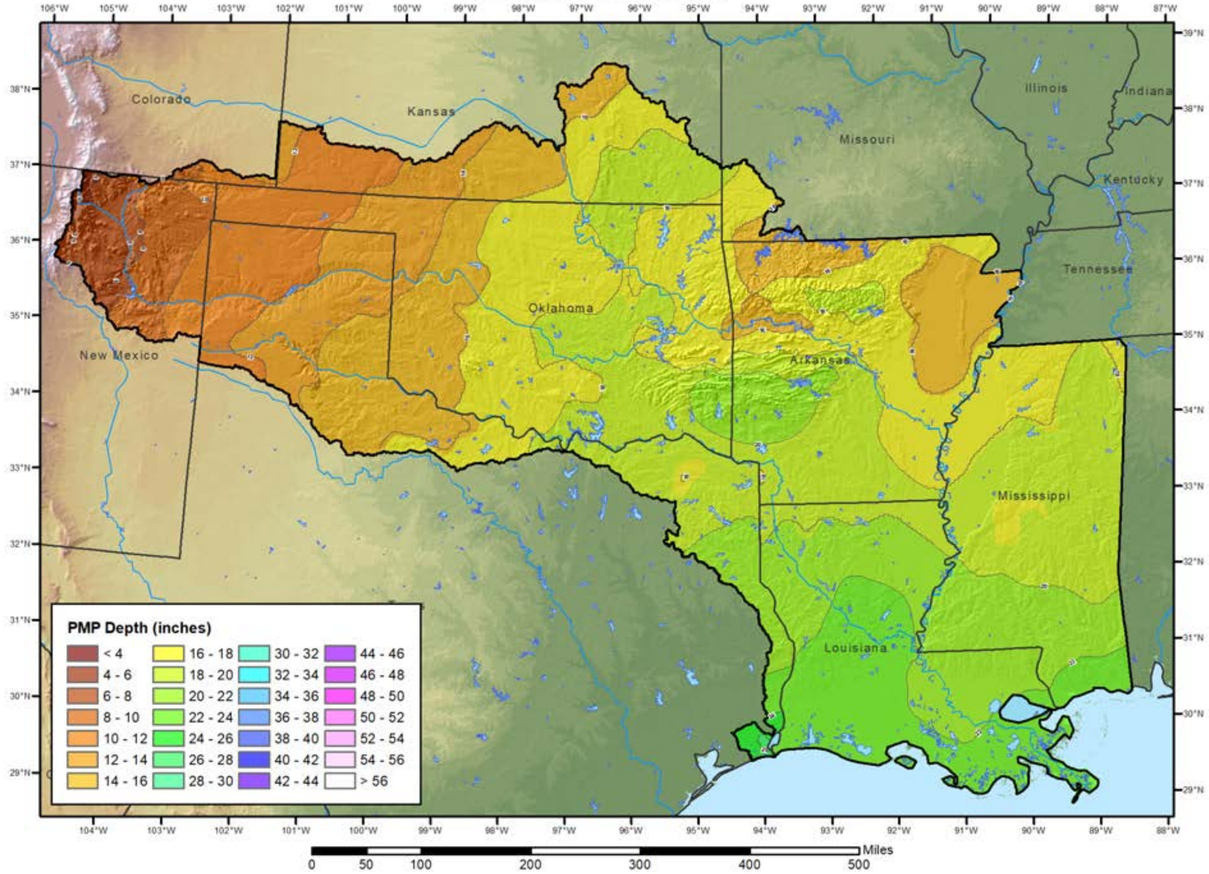
10,000 mi$^2$ 24-hour - General Storm PMP (in.)
AR-LA-MS-OK PMP Study



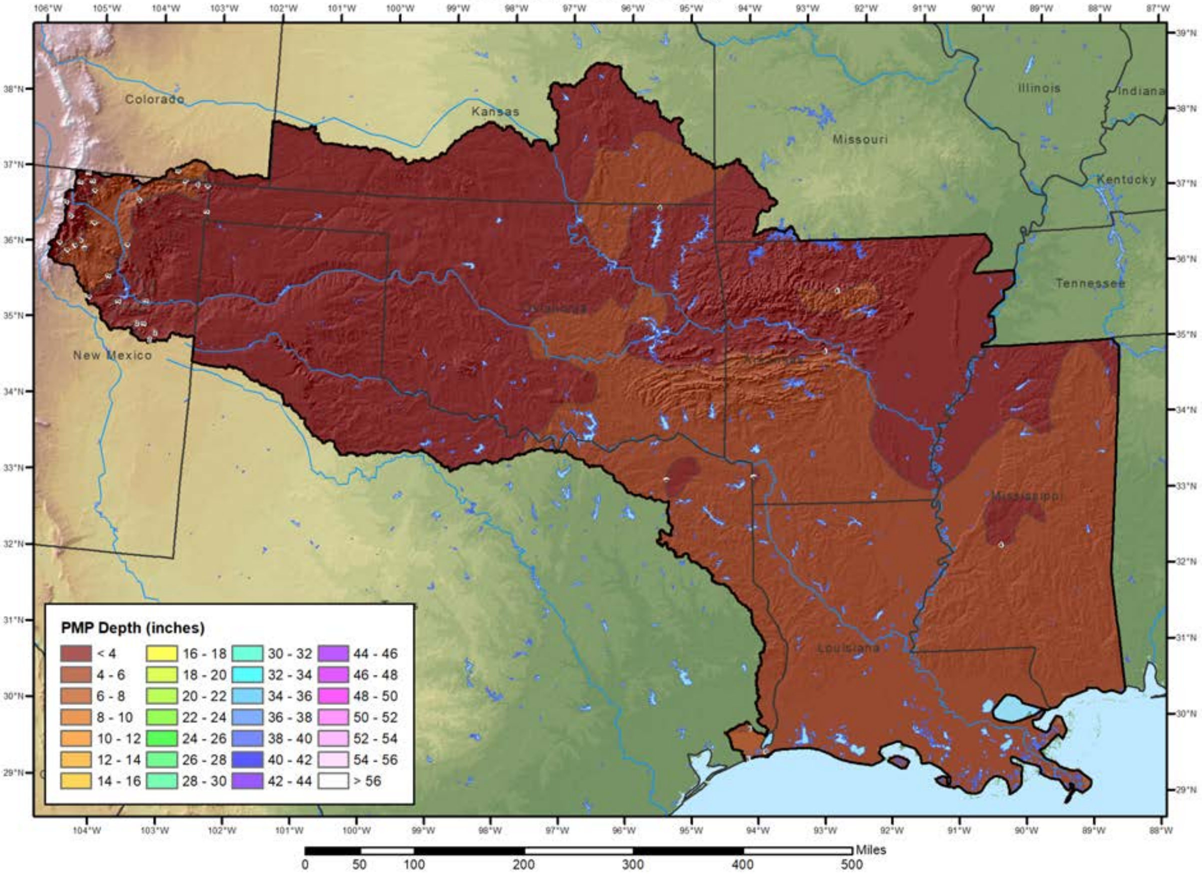
10,000 mi$^2$ 48-hour - General Storm PMP (in.)
AR-LA-MS-OK PMP Study



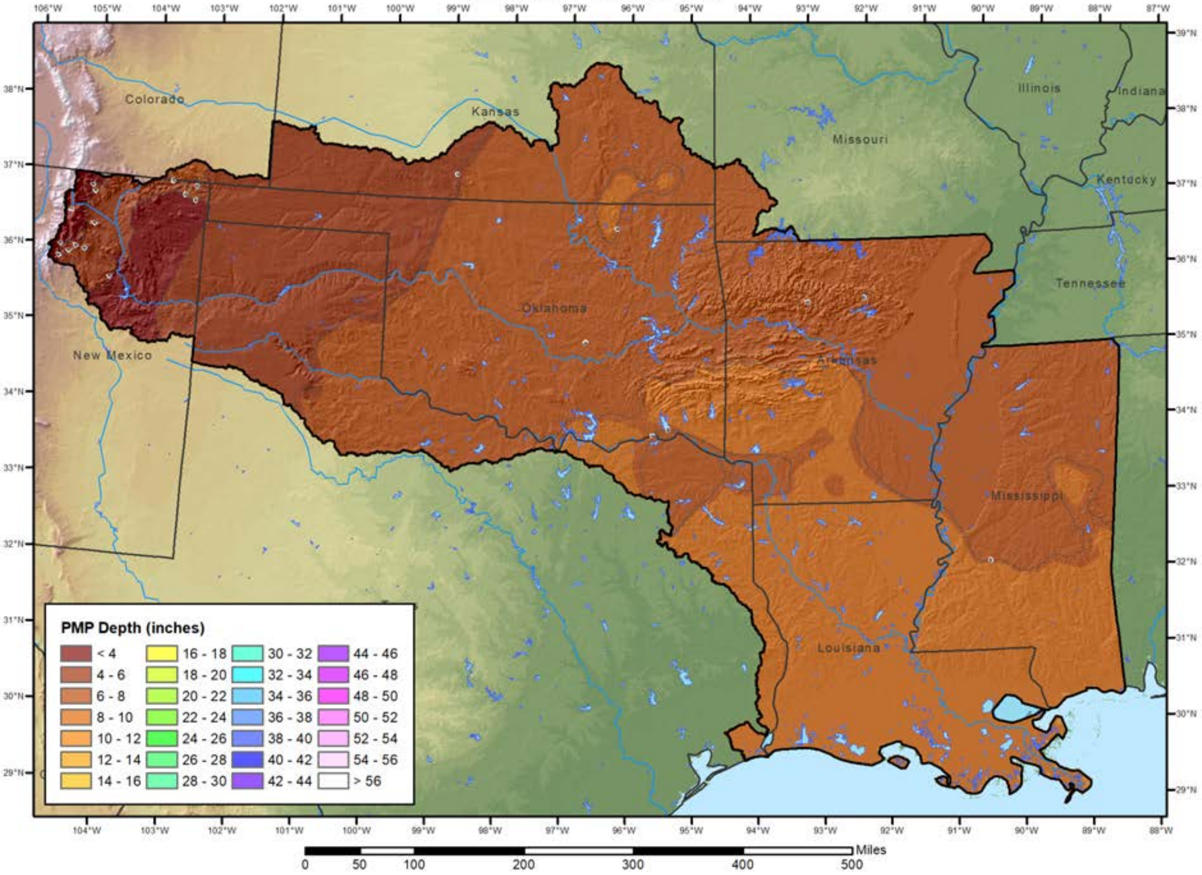
10,000 mi$^2$ 72-hour - General Storm PMP (in.)
AR-LA-MS-OK PMP Study



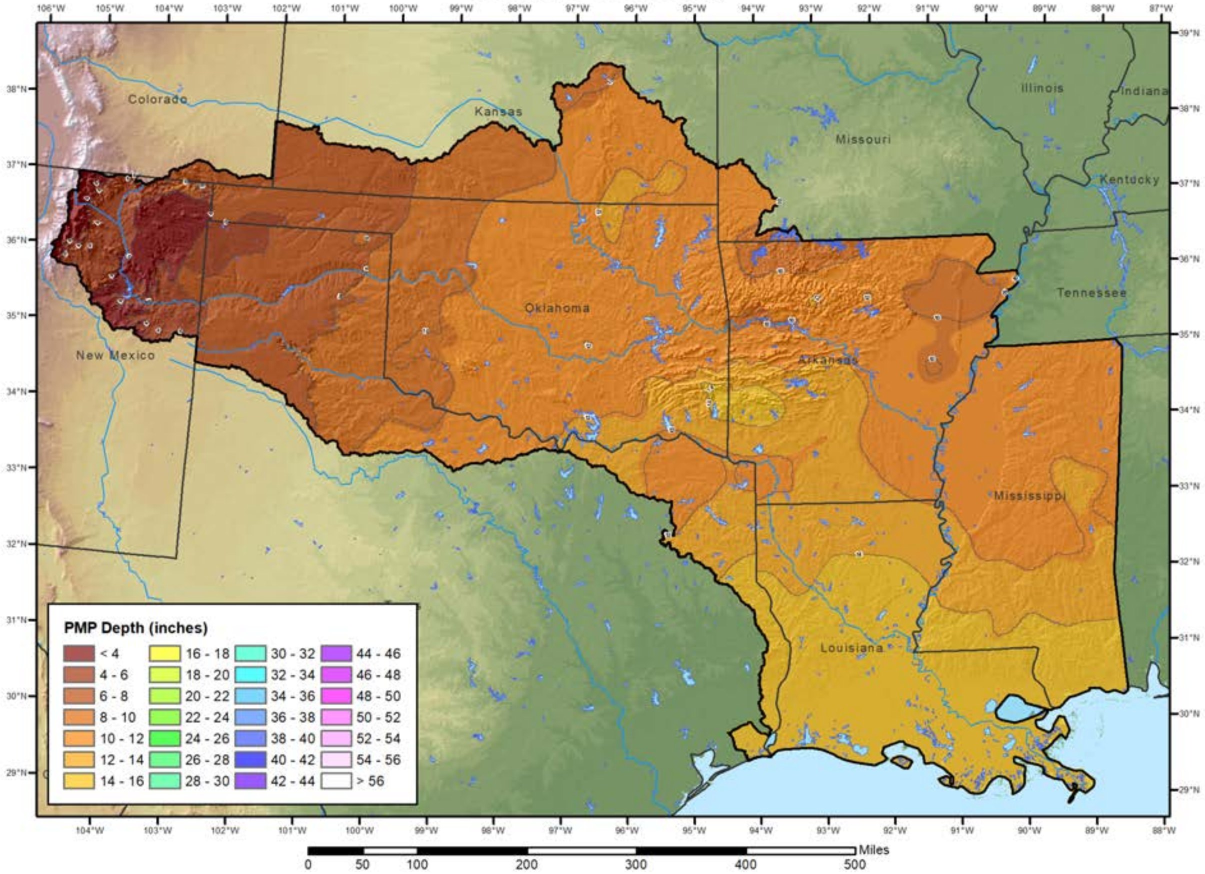
20,000 $mi^2$ 6-hour - General Storm PMP (in.)
AR-LA-MS-OK PMP Study



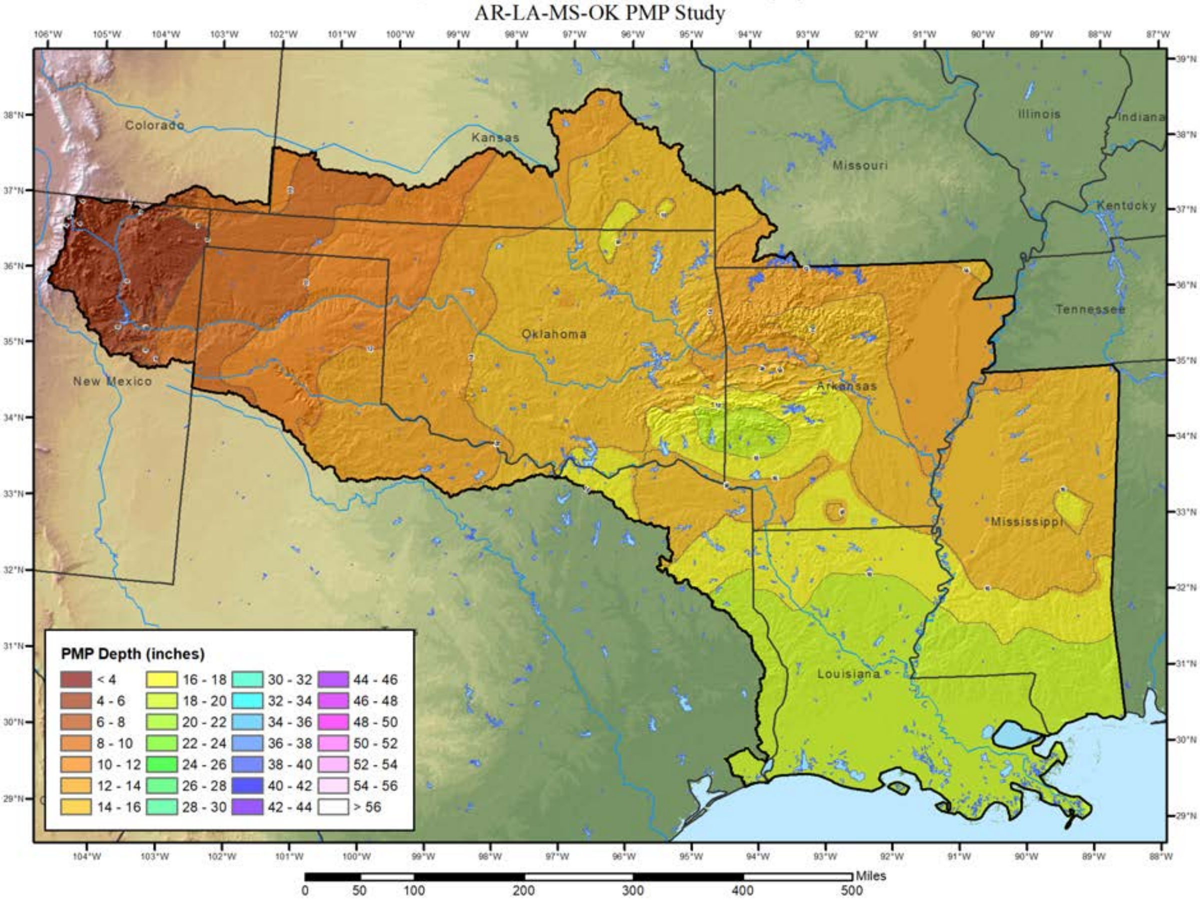
20,000 mi$^2$ 12-hour - General Storm PMP (in.)
AR-LA-MS-OK PMP Study



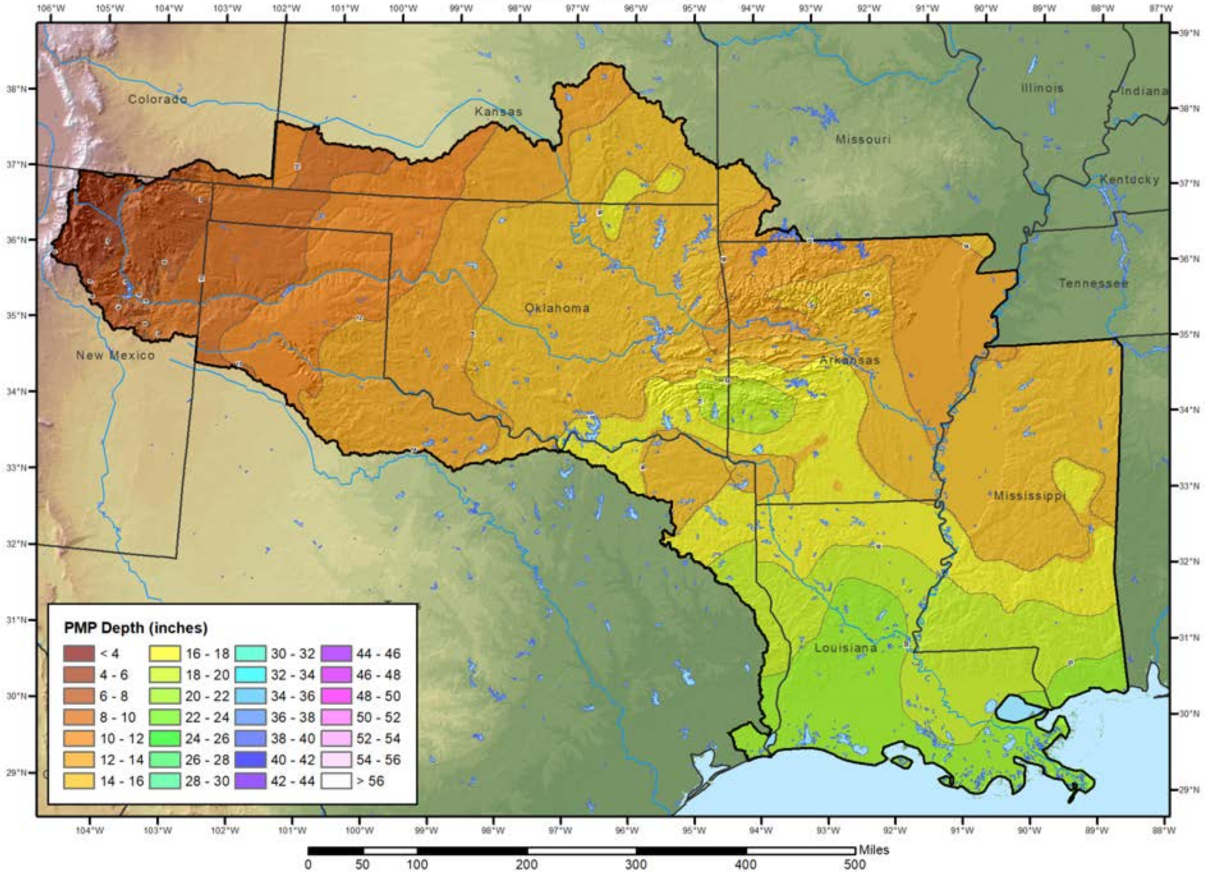
20,000 mi$^2$ 24-hour - General Storm PMP (in.)
AR-LA-MS-OK PMP Study



20,000 mi$^2$ 48-hour - General Storm PMP (in.)
AR-LA-MS-OK PMP Study

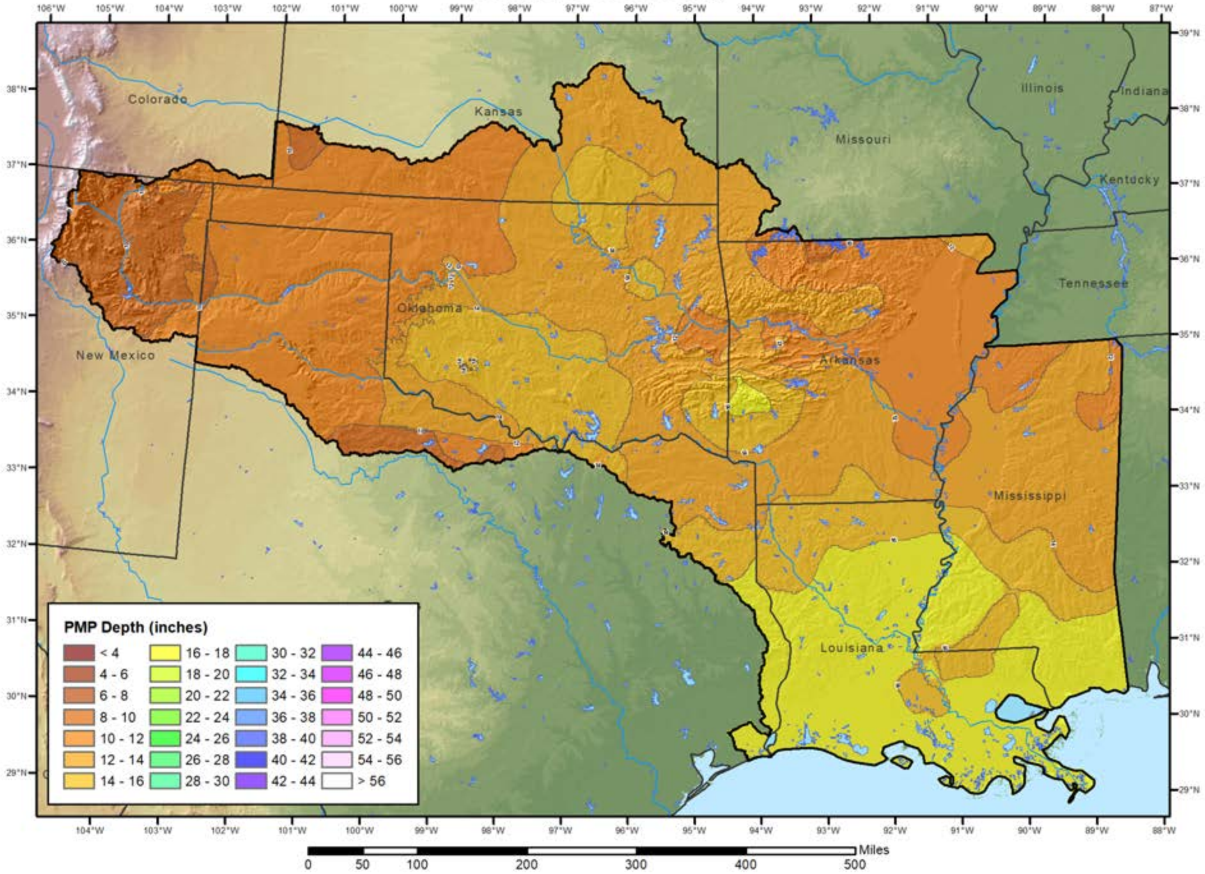


20,000 mi$^2$ 72-hour - General Storm PMP (in.)
AR-LA-MS-OK PMP Study

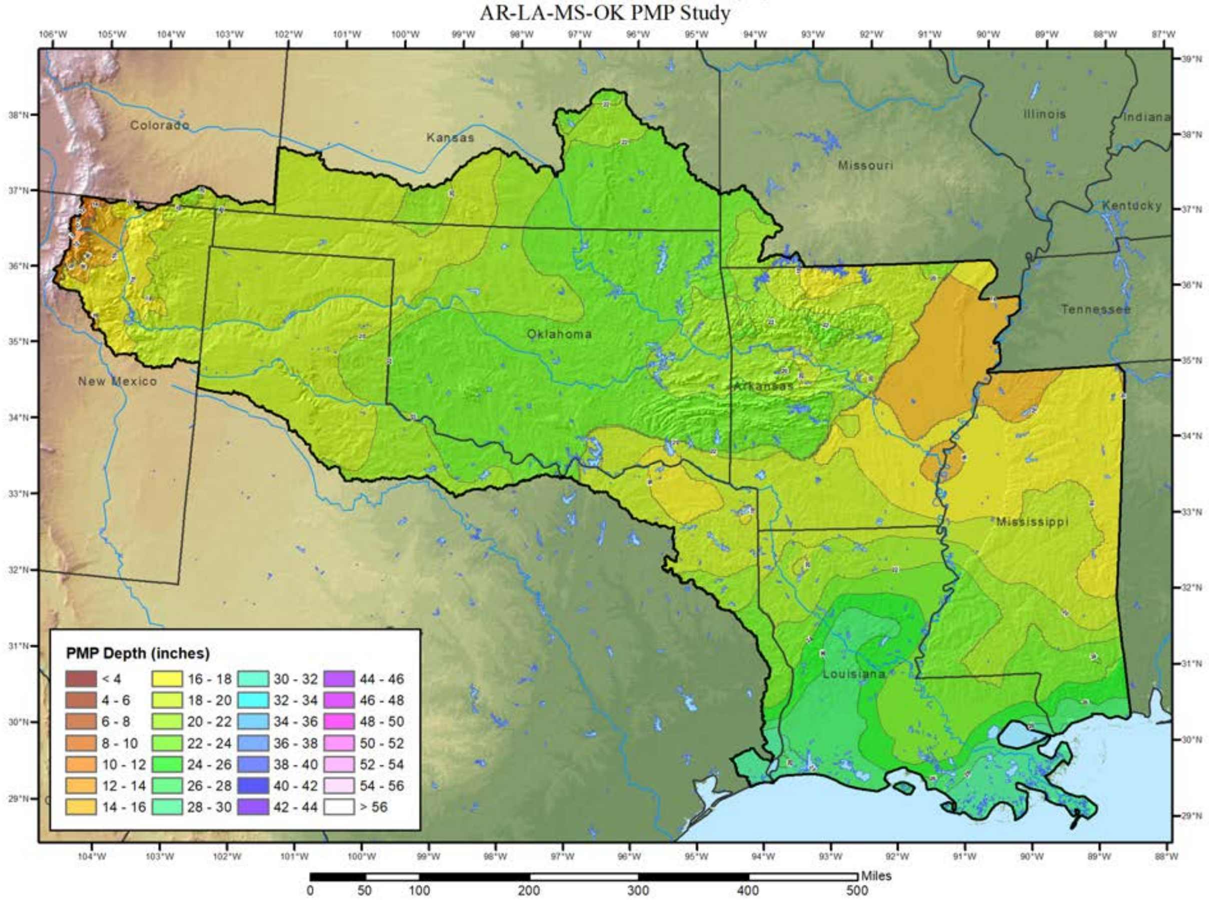


# Local Storms

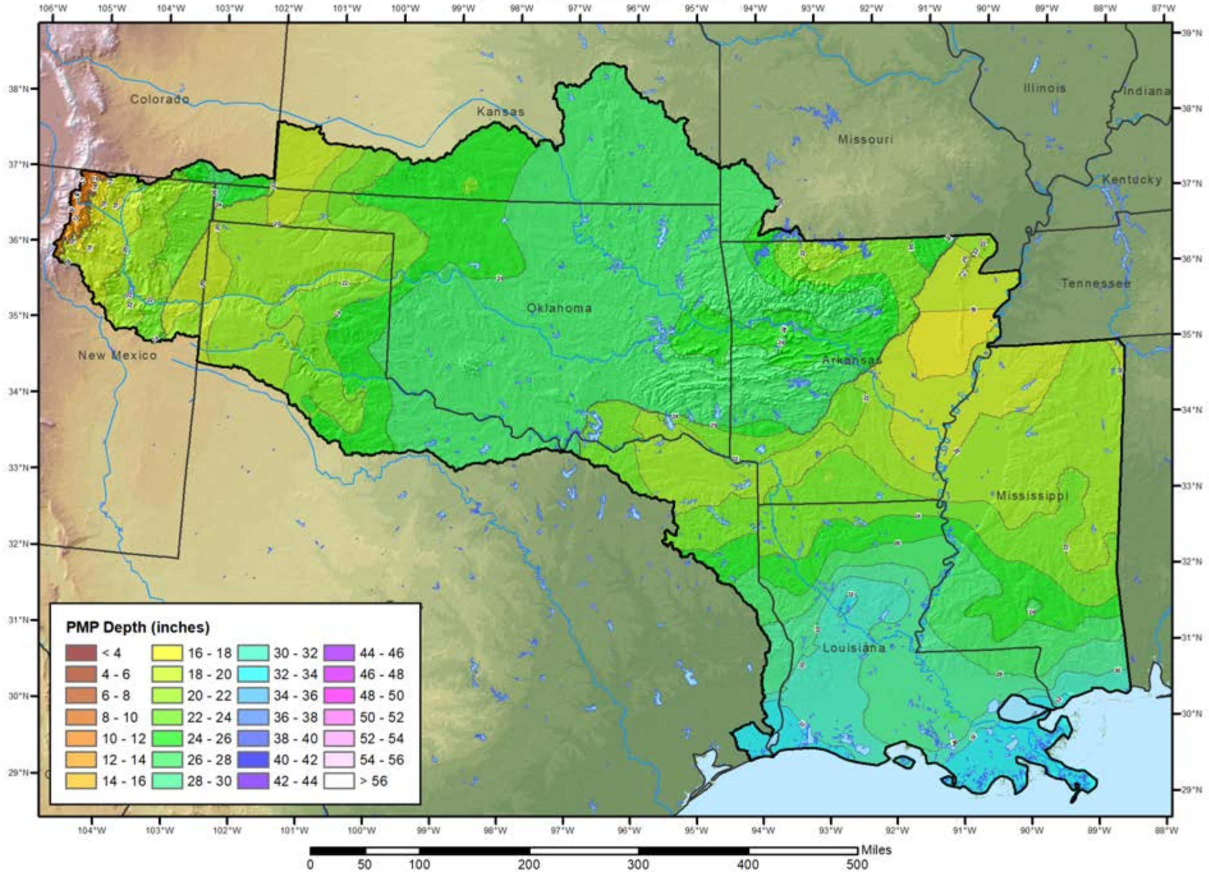
1 mi$^2$ 1-hour - Local Storm PMP (in.)
AR-LA-MS-OK PMP Study



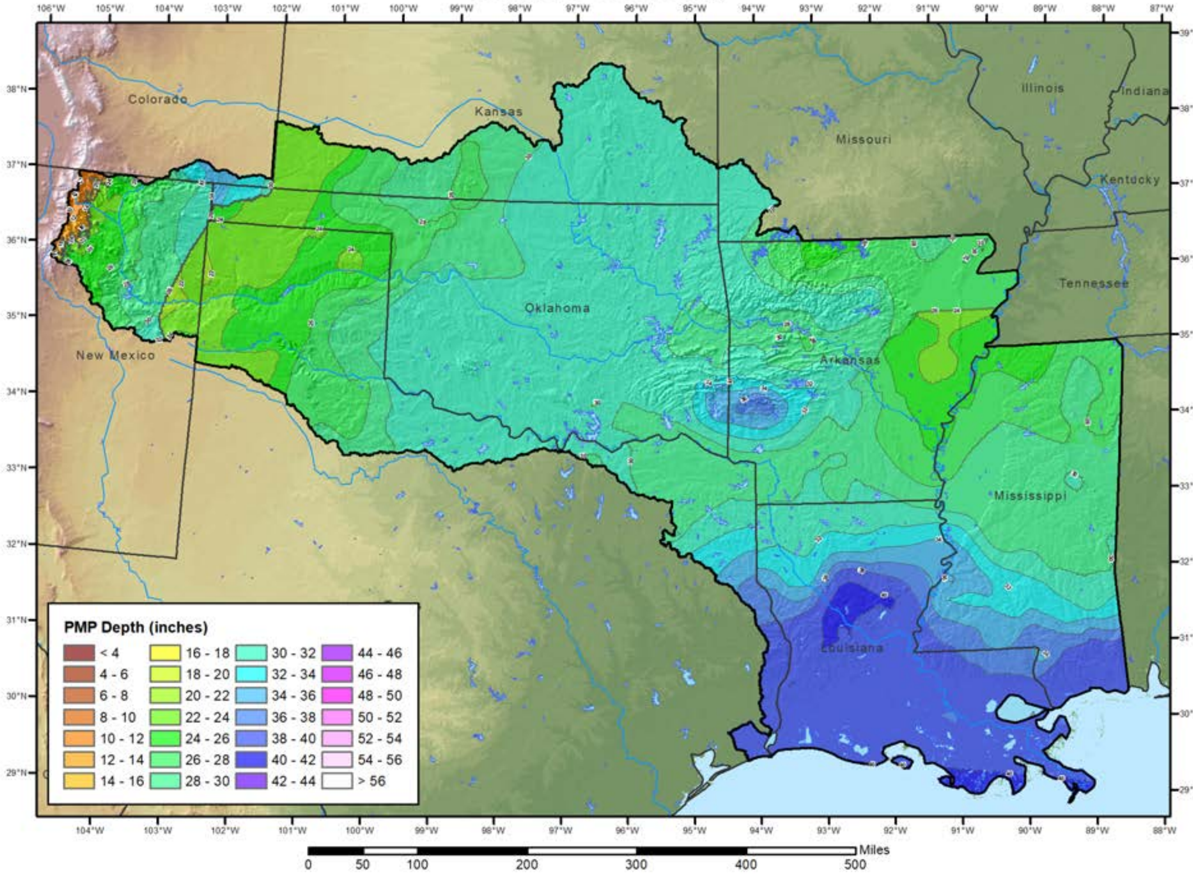
1 mi$^2$ 3-hour - Local Storm PMP (in.)
AR-LA-MS-OK PMP Study



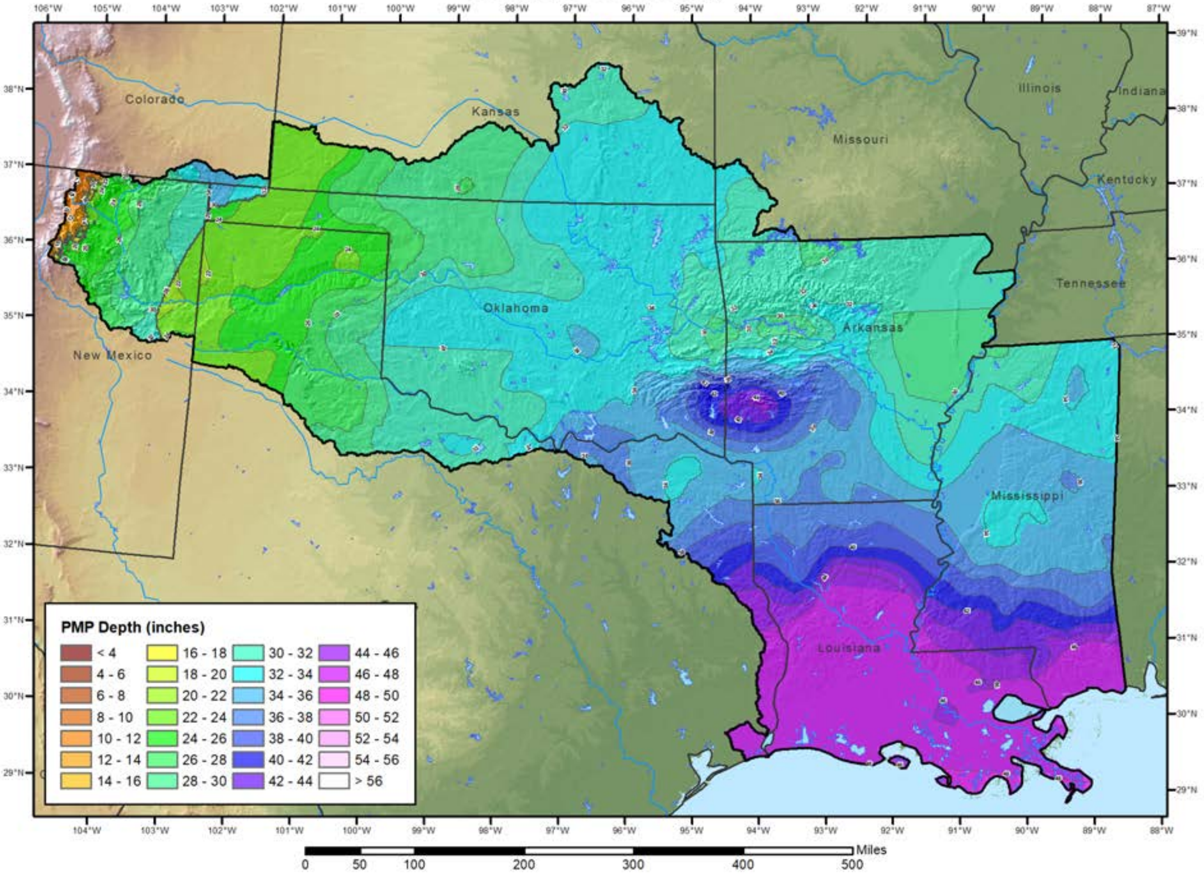
1 mi$^2$ 6-hour - Local Storm PMP (in.)
AR-LA-MS-OK PMP Study



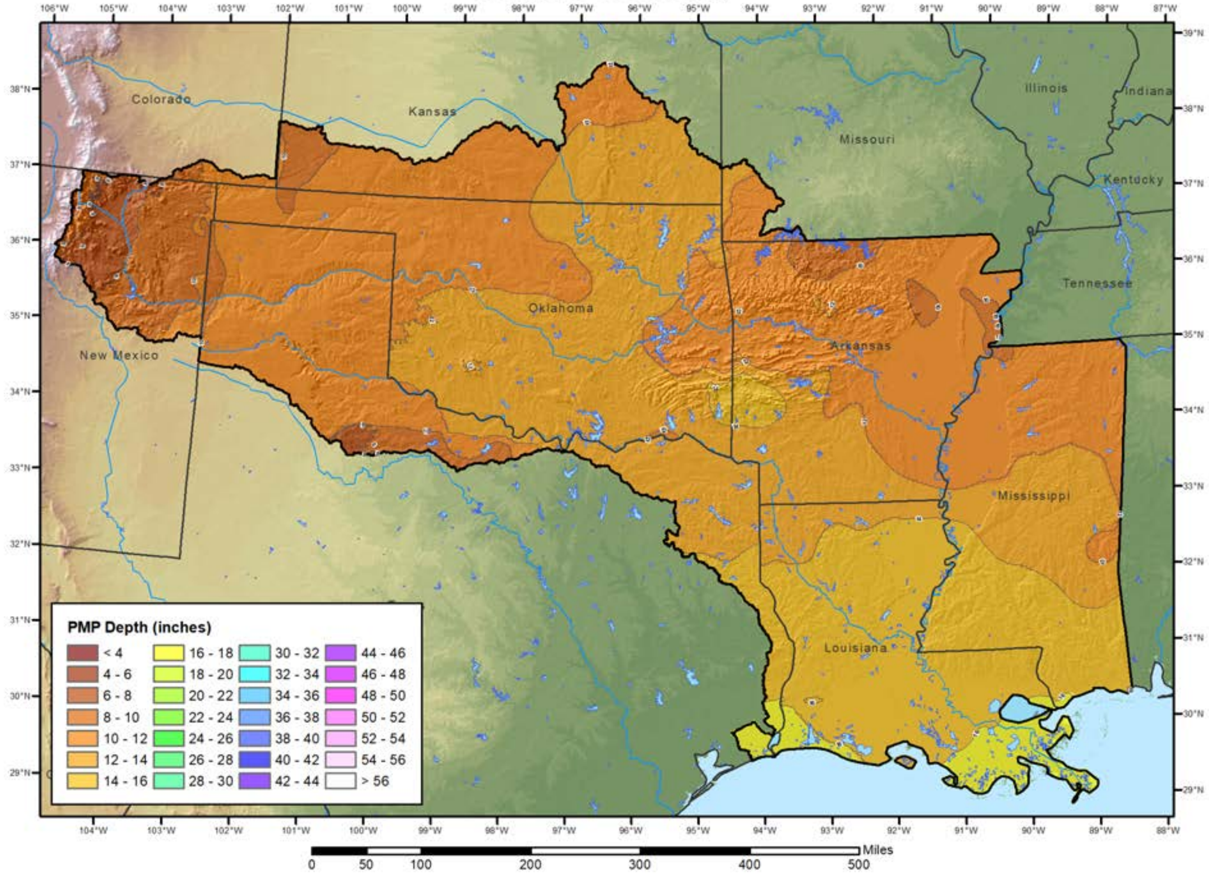
1 mi$^2$ 12-hour - Local Storm PMP (in.)
AR-LA-MS-OK PMP Study



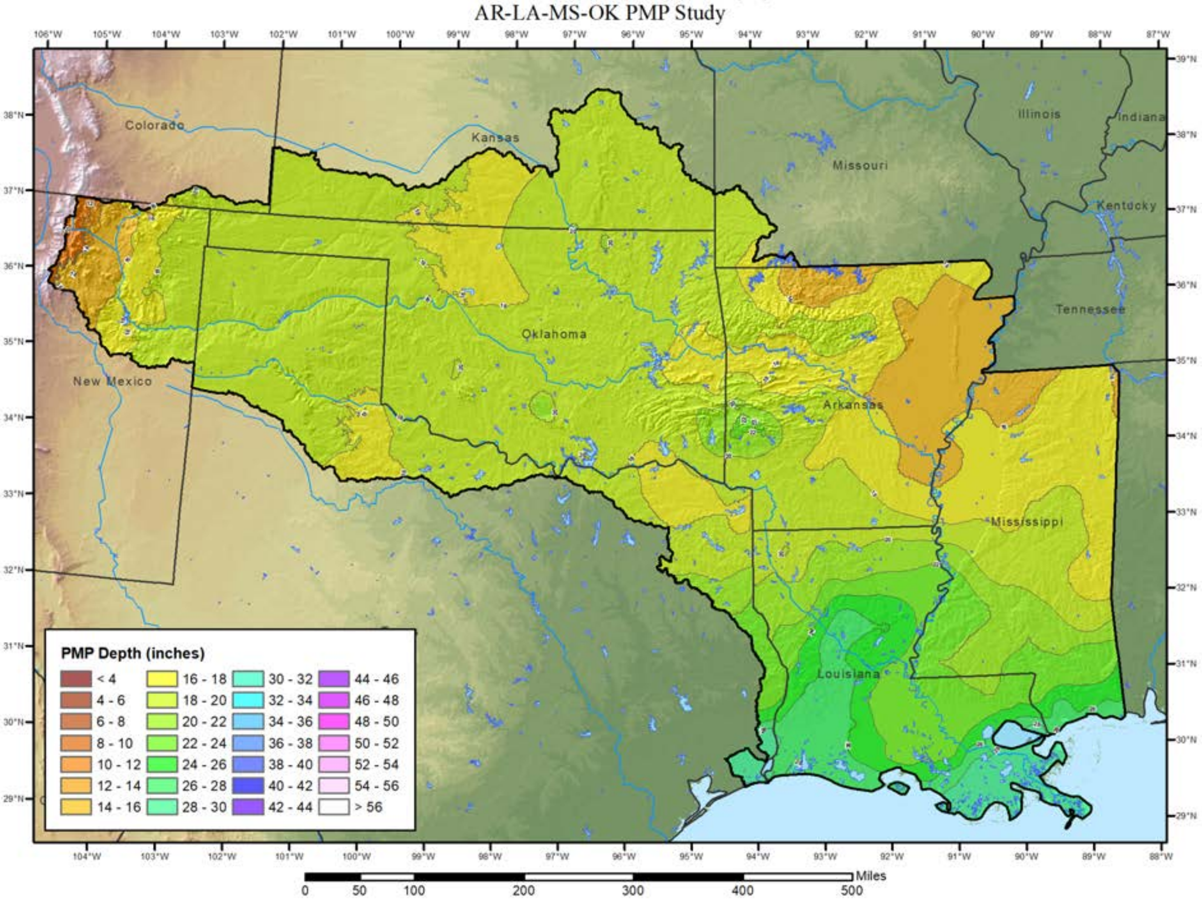
1 mi$^2$ 24-hour - Local Storm PMP (in.)
AR-LA-MS-OK PMP Study



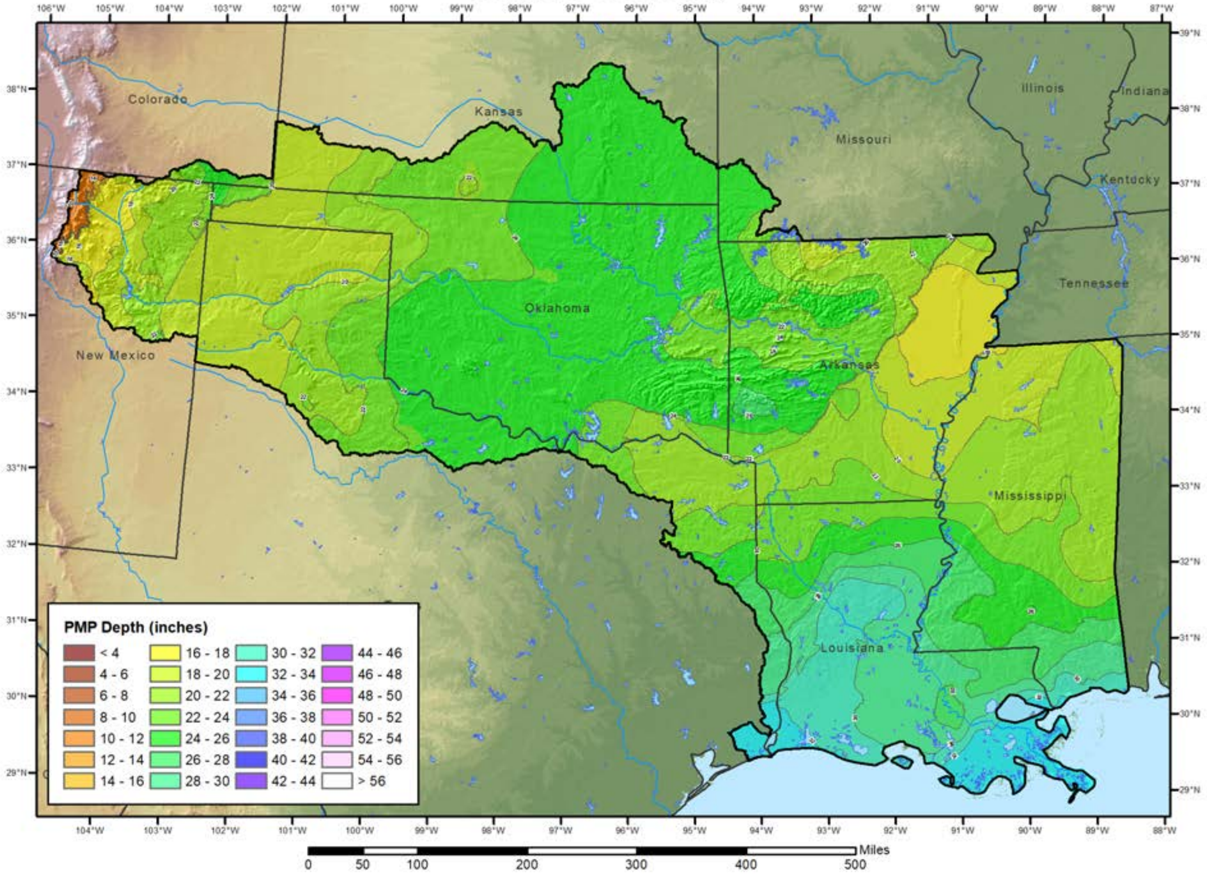
10 mi$^2$ 1-hour - Local Storm PMP (in.)
AR-LA-MS-OK PMP Study



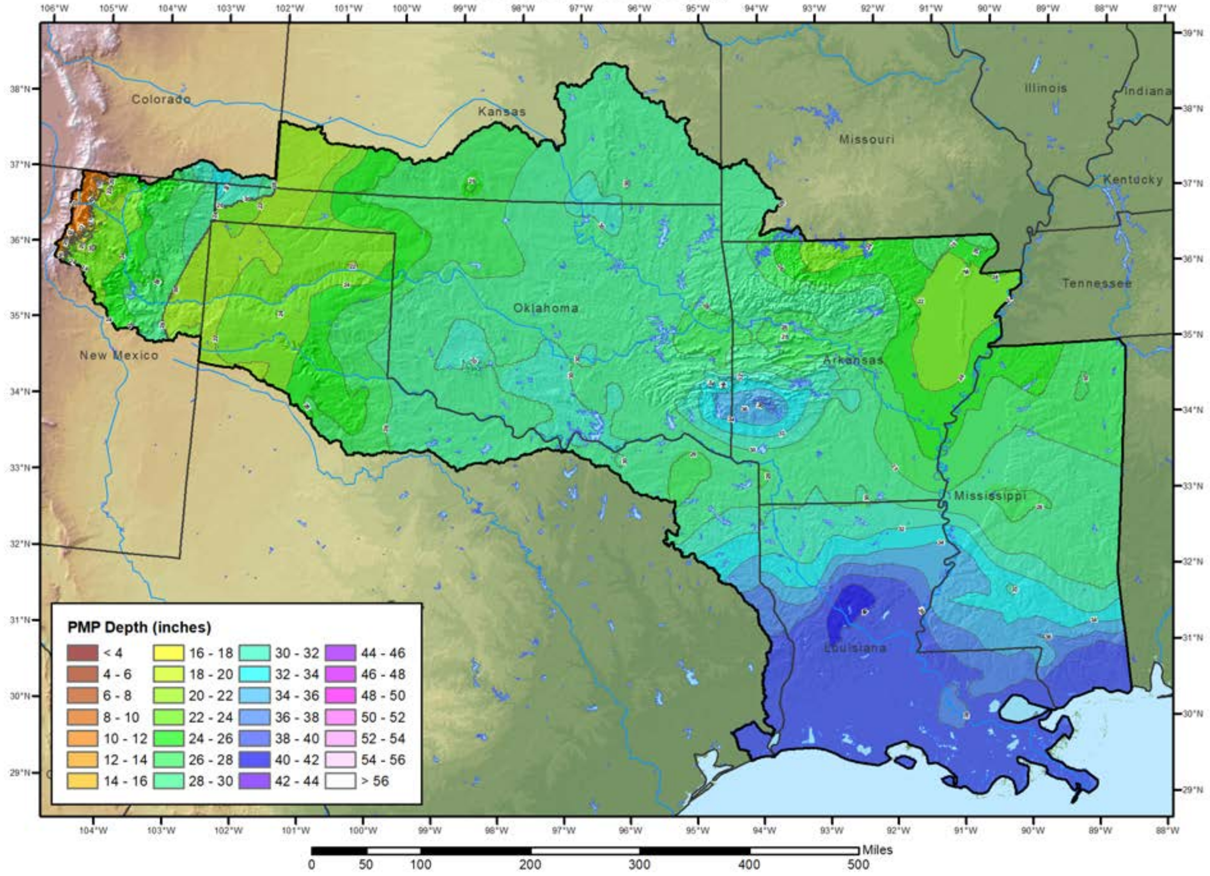
10 mi$^2$ 3-hour - Local Storm PMP (in.)
AR-LA-MS-OK PMP Study



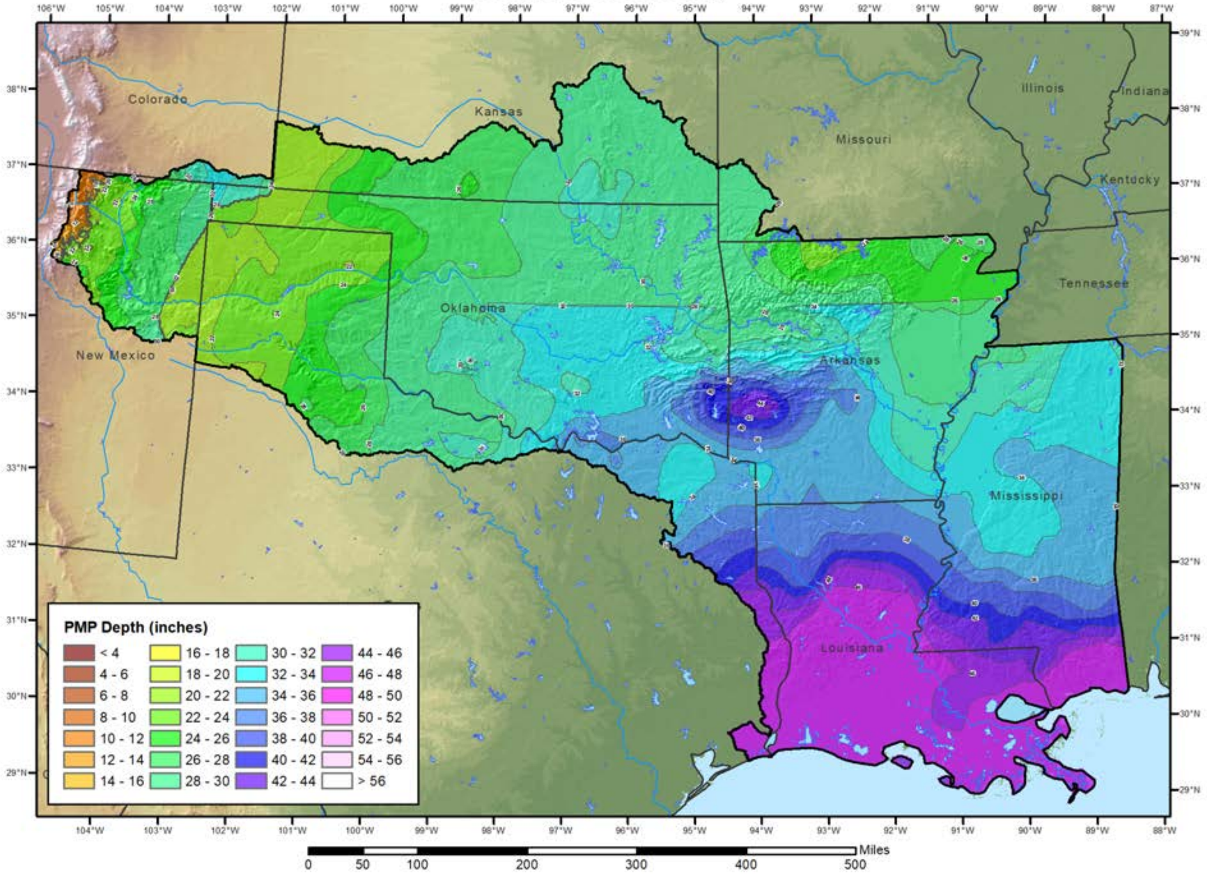
10 mi$^2$ 6-hour - Local Storm PMP (in.)
AR-LA-MS-OK PMP Study



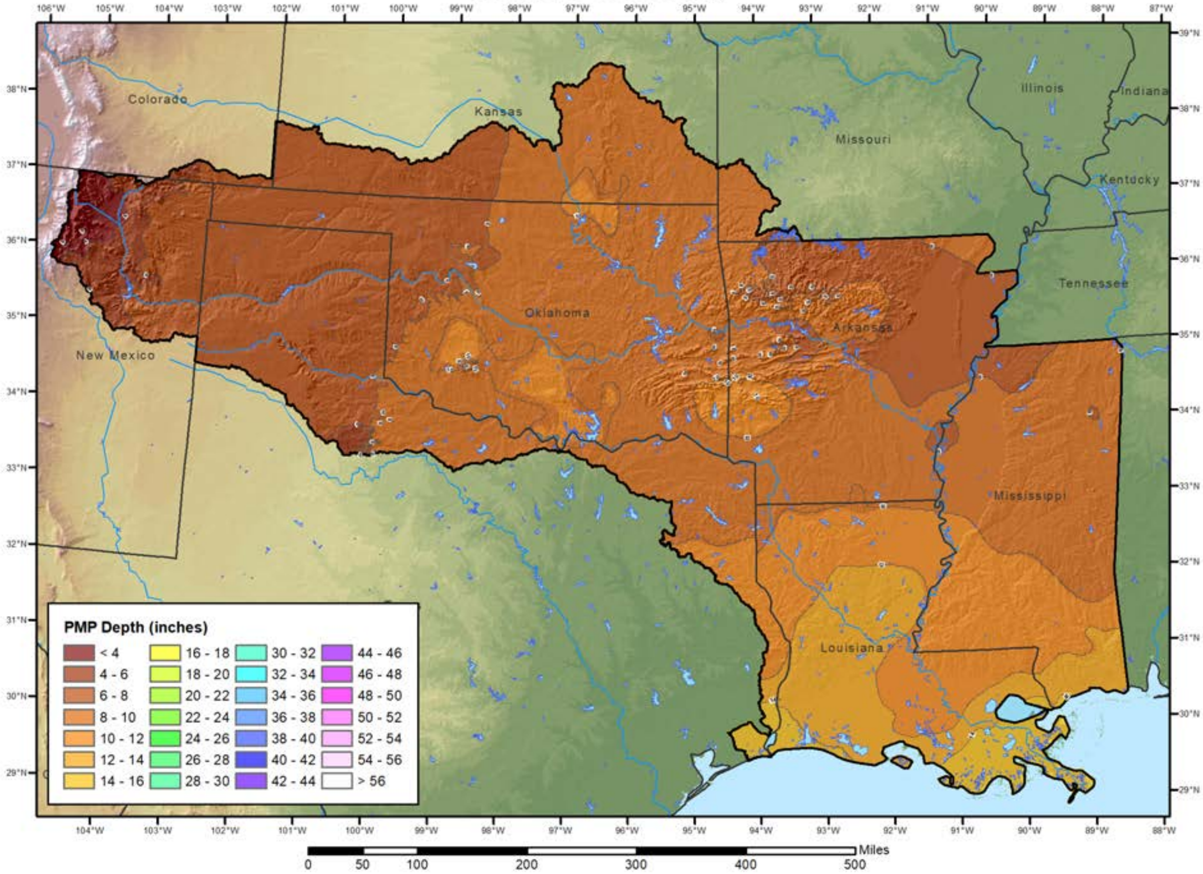
10 mi$^2$ 12-hour - Local Storm PMP (in.)
AR-LA-MS-OK PMP Study



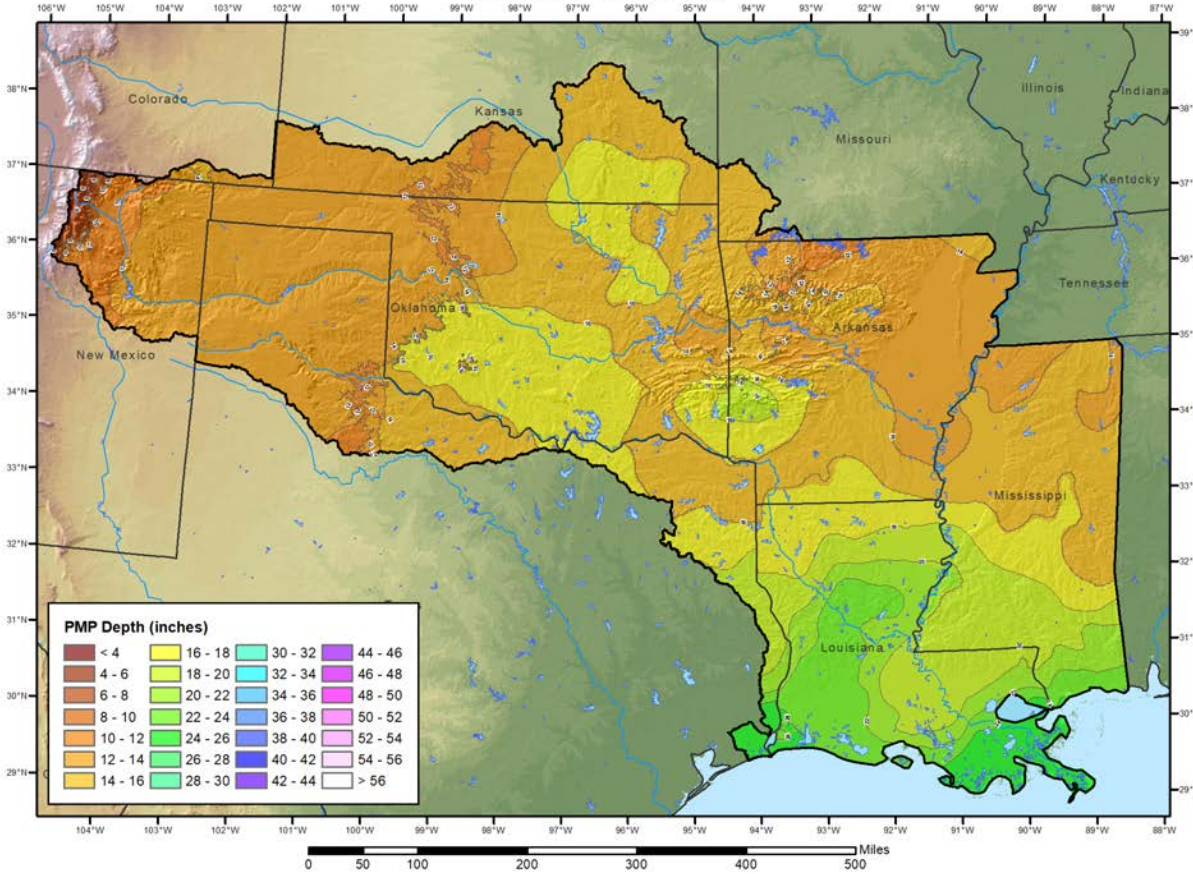
10 mi$^2$ 24-hour - Local Storm PMP (in.)
AR-LA-MS-OK PMP Study



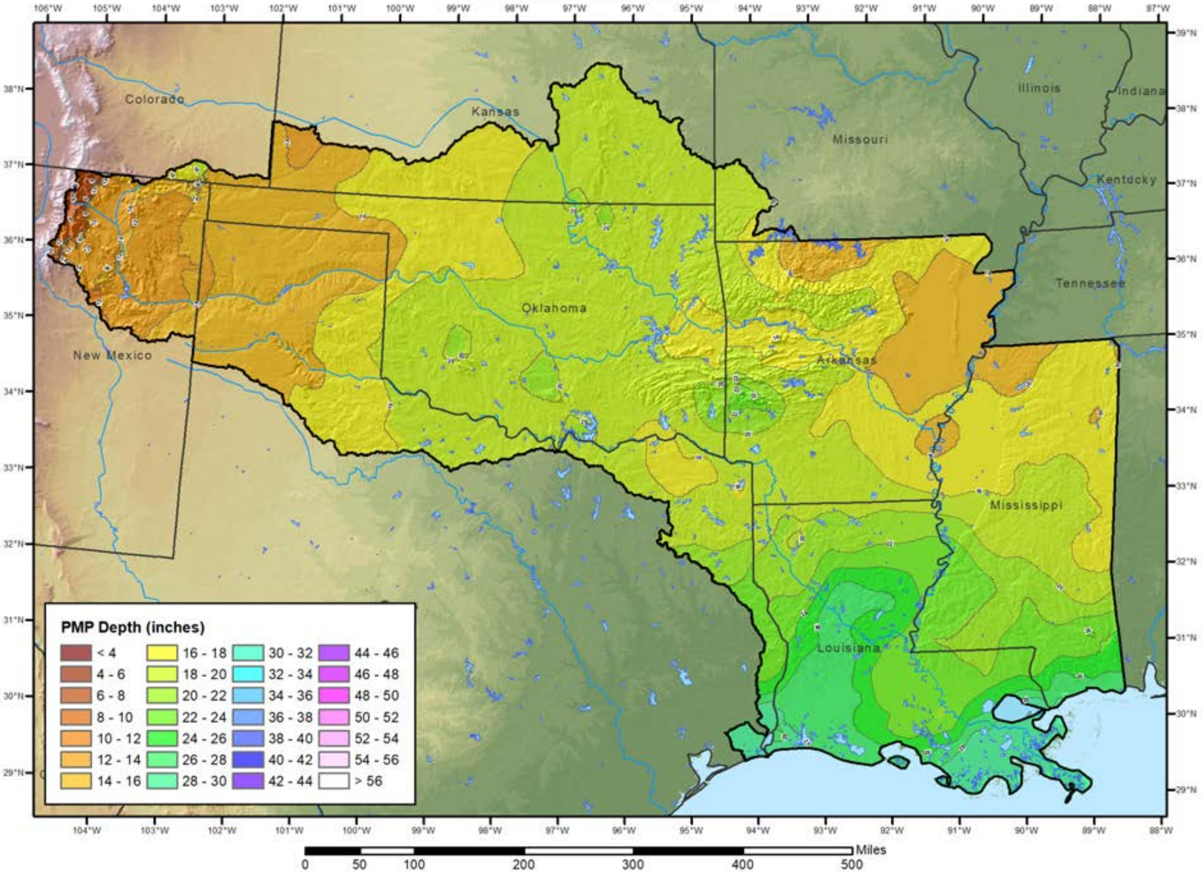
200 mi$^2$ 1-hour - Local Storm PMP (in.)
AR-LA-MS-OK PMP Study



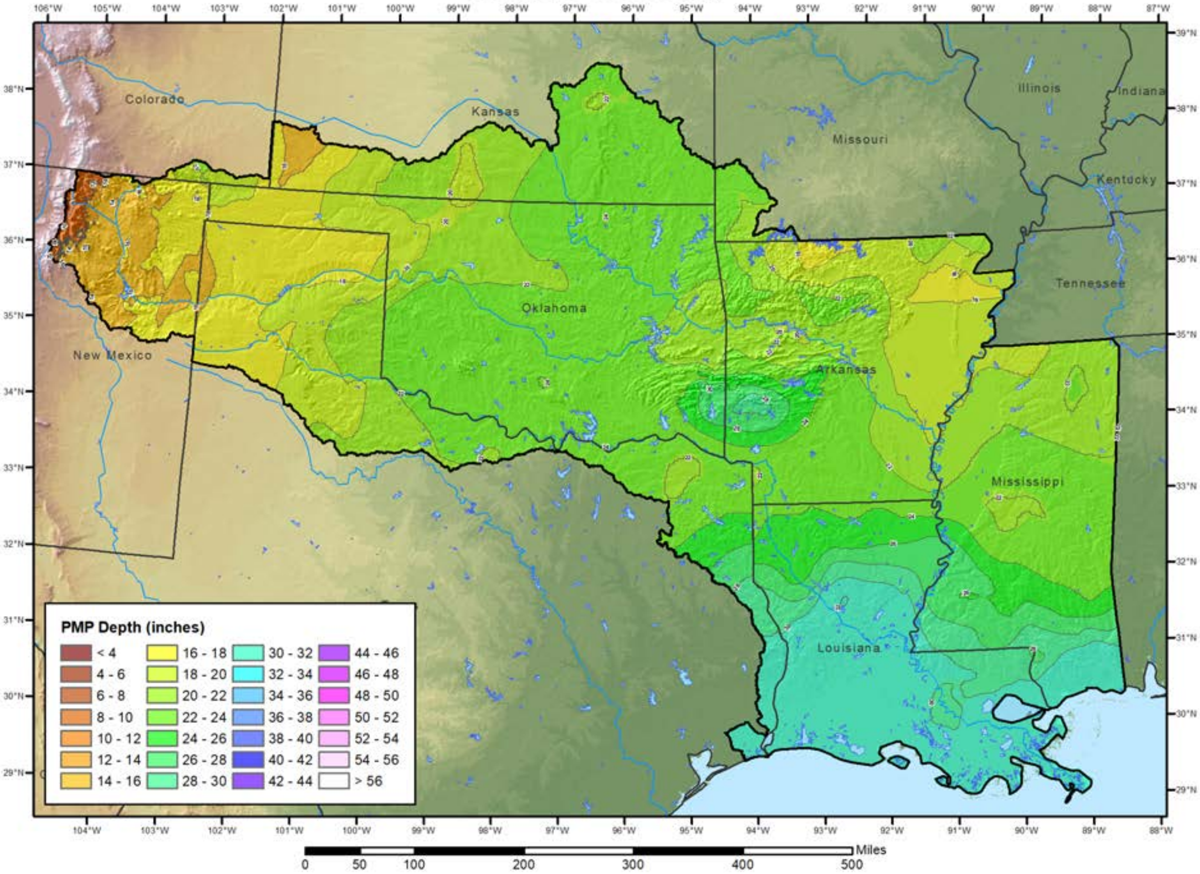
200 mi$^2$ 3-hour - Local Storm PMP (in.)
AR-LA-MS-OK PMP Study



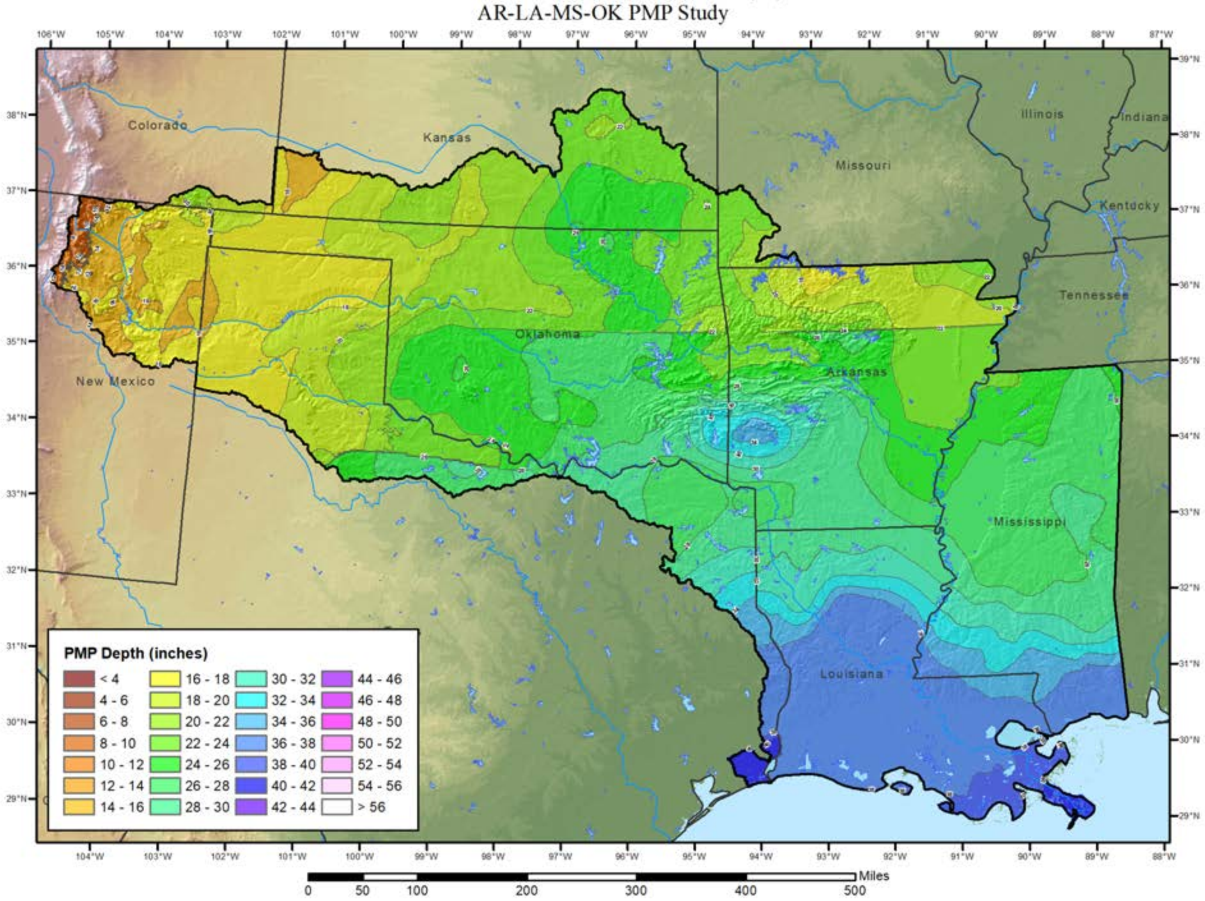
200 mi$^2$ 6-hour - Local Storm PMP (in.)
AR-LA-MS-OK PMP Study



200 mi$^2$ 12-hour - Local Storm PMP (in.)
AR-LA-MS-OK PMP Study

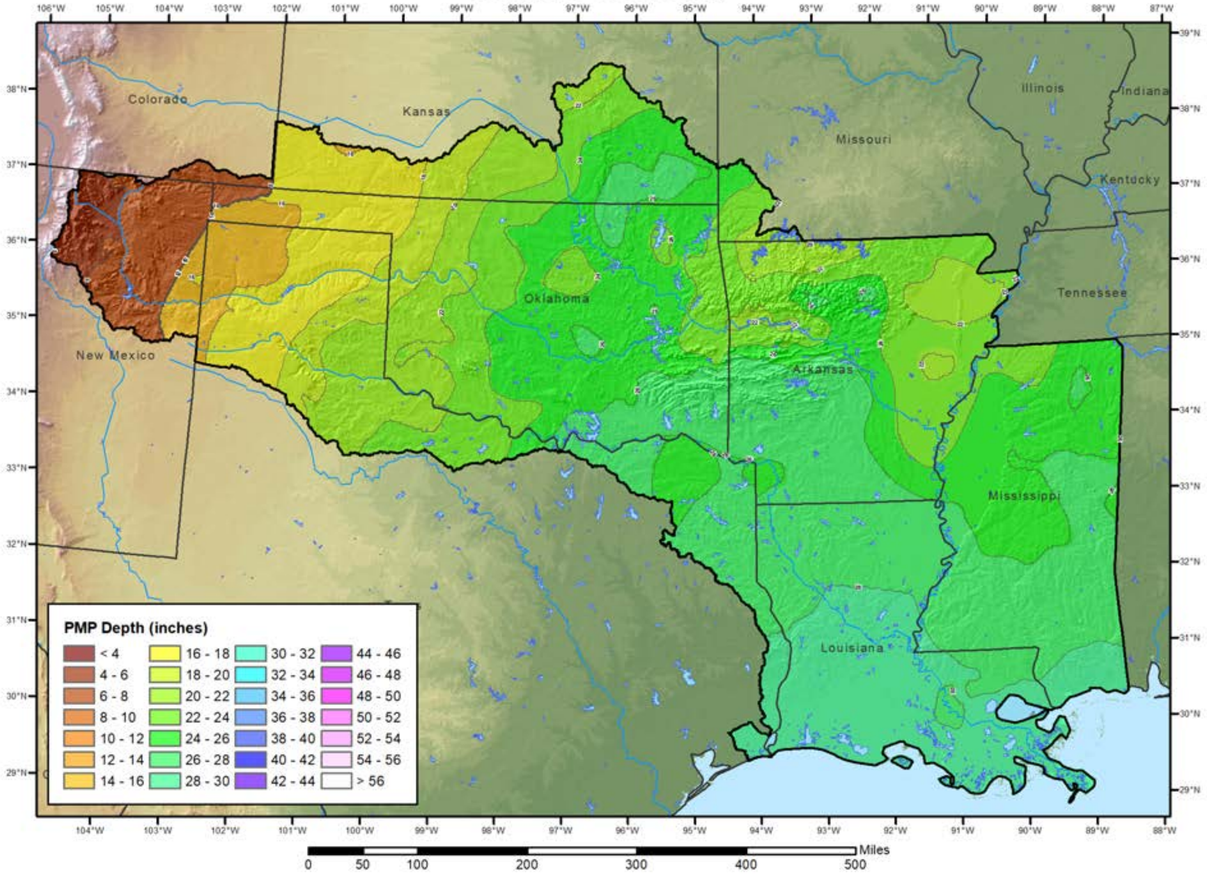


200 mi$^2$ 24-hour - Local Storm PMP (in.)
AR-LA-MS-OK PMP Study

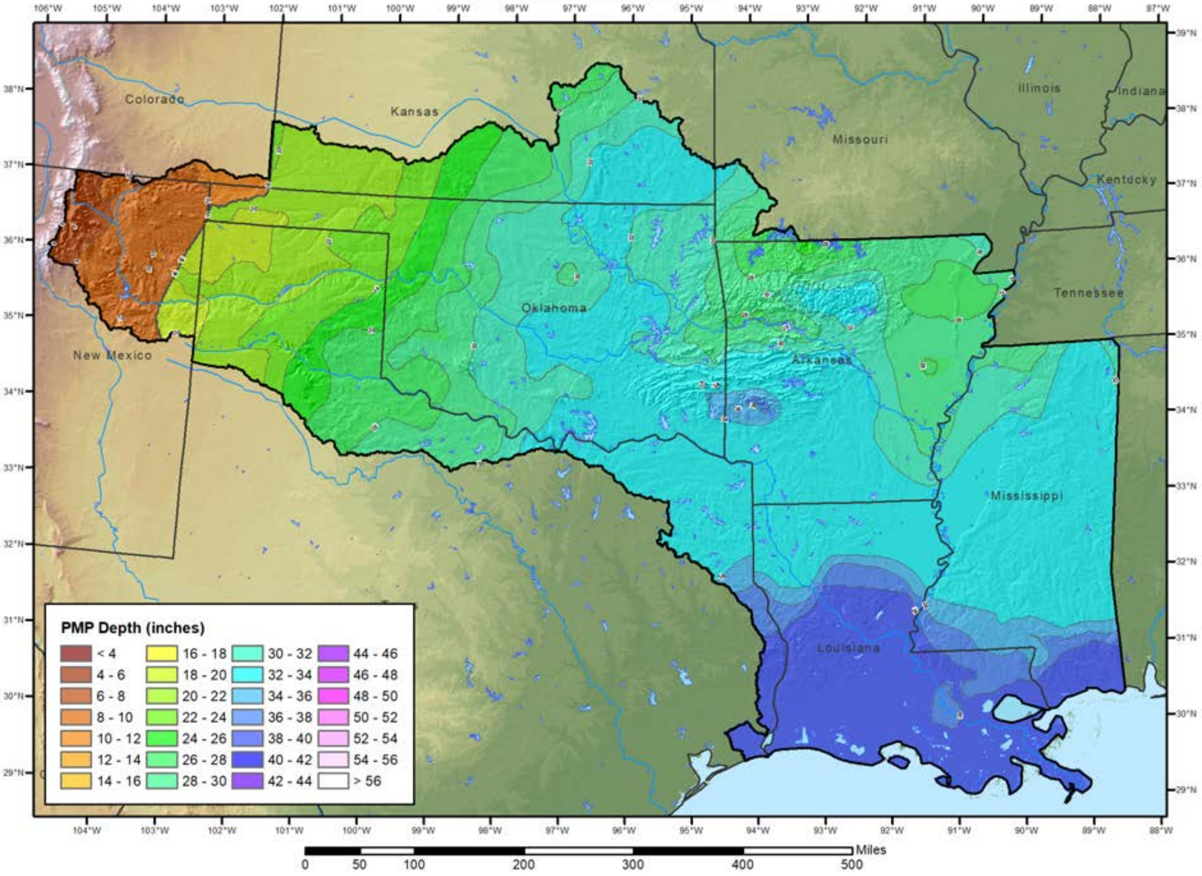


# Tropical Storms

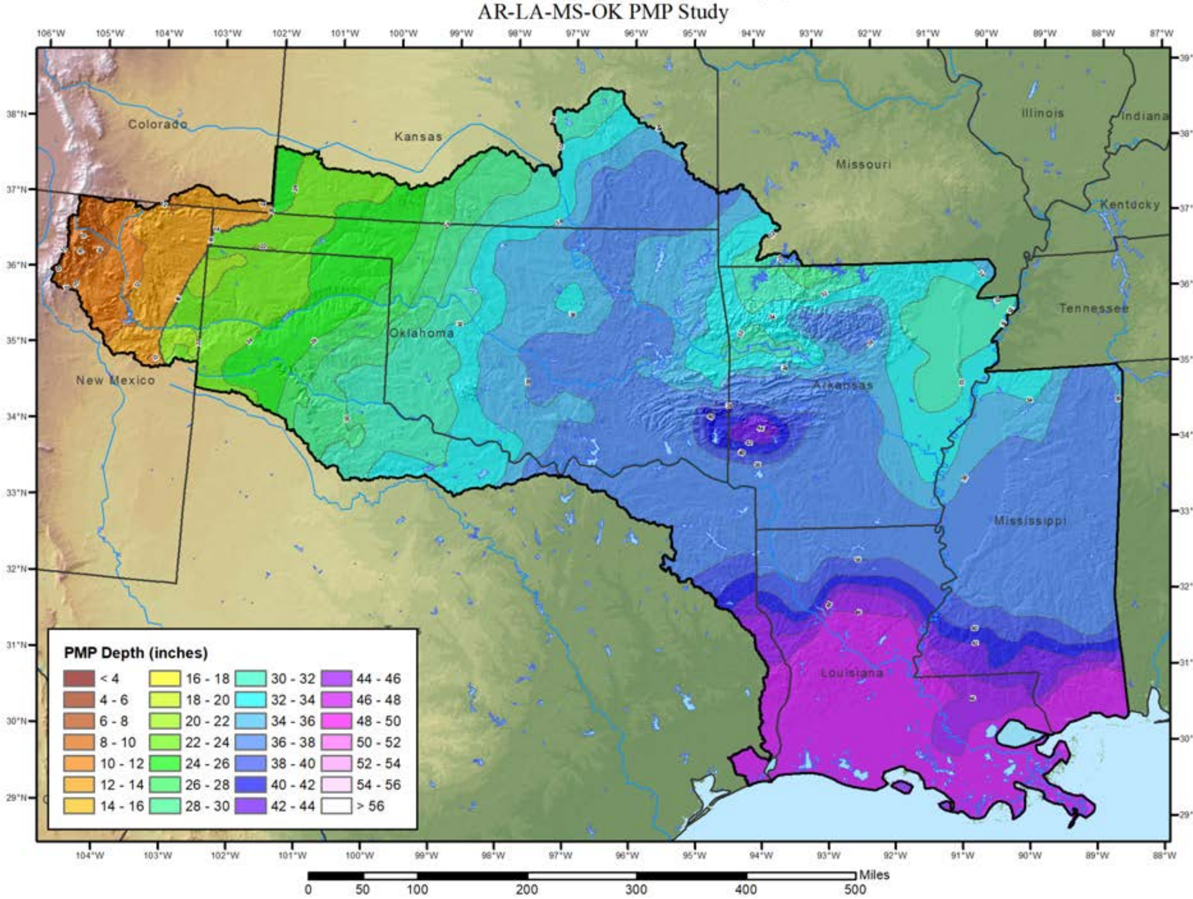
10 mi$^2$ 6-hour - Tropical Storm PMP (in.)
AR-LA-MS-OK PMP Study



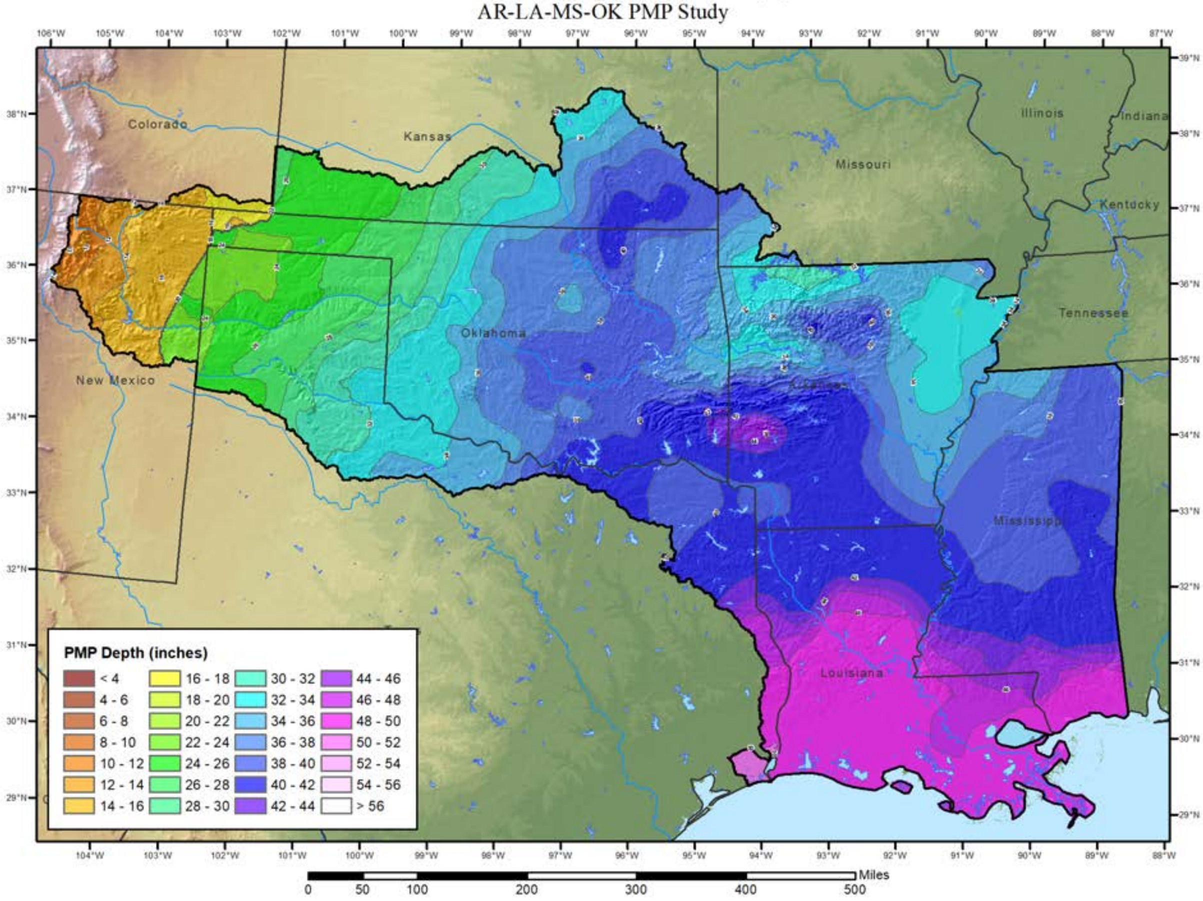
10 mi$^2$ 12-hour - Tropical Storm PMP (in.)
AR-LA-MS-OK PMP Study



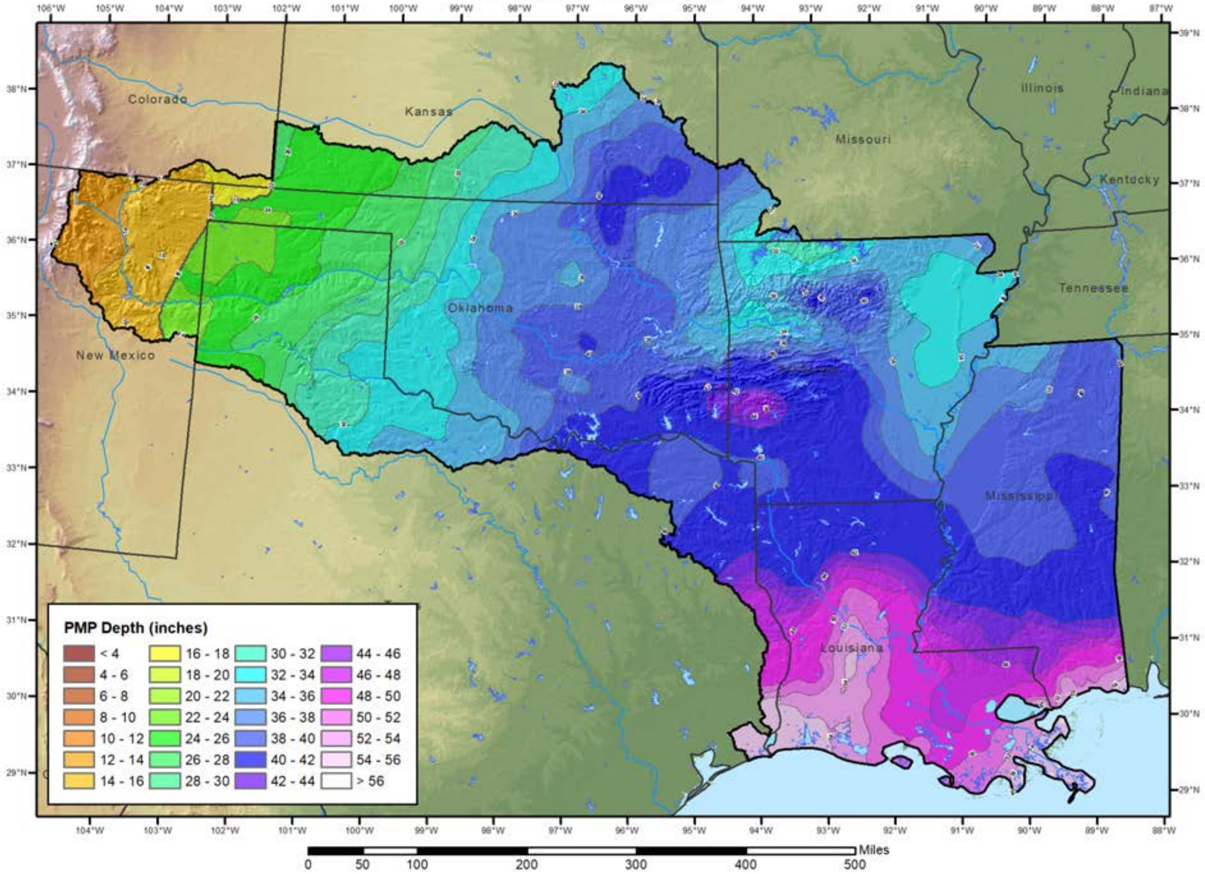
10 mi$^2$ 24-hour - Tropical Storm PMP (in.)
AR-LA-MS-OK PMP Study



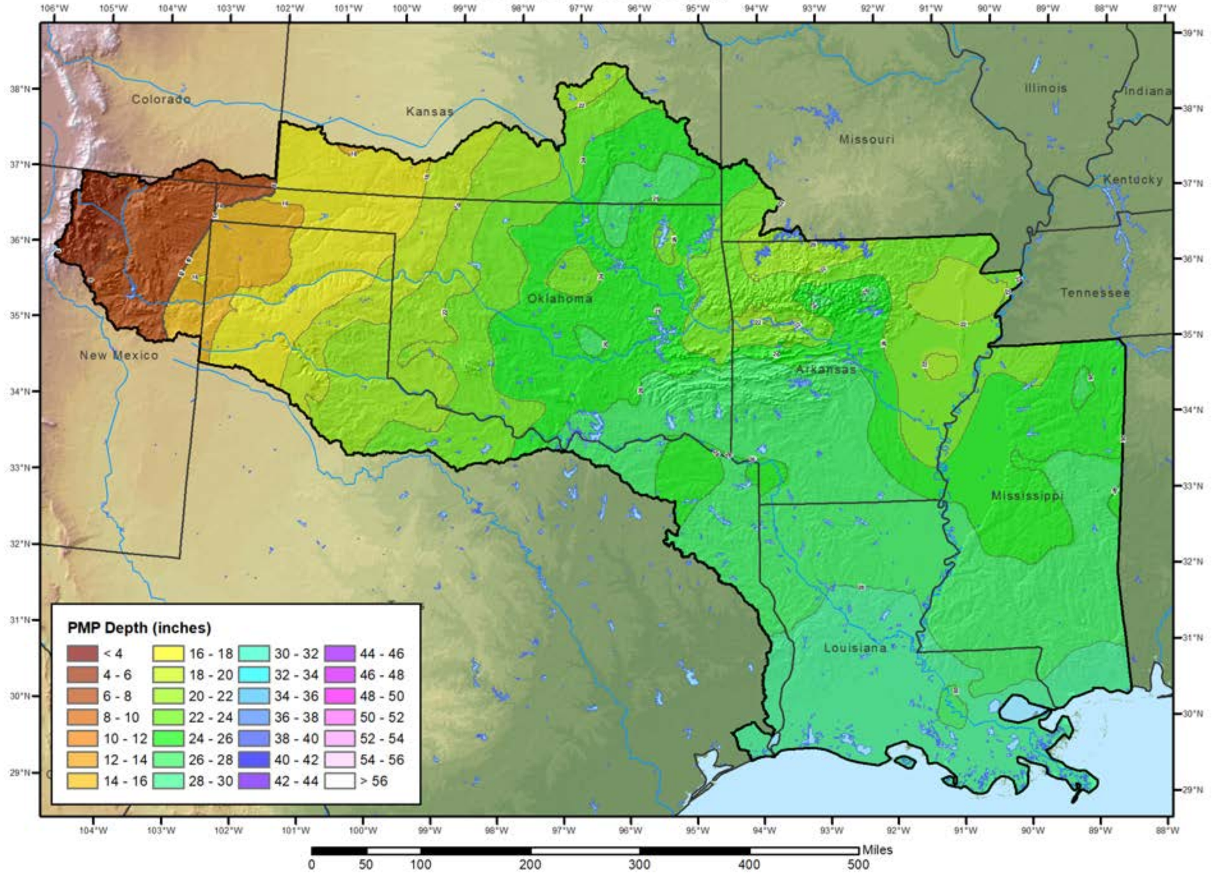
10 mi$^2$ 48-hour - Tropical Storm PMP (in.)
AR-LA-MS-OK PMP Study



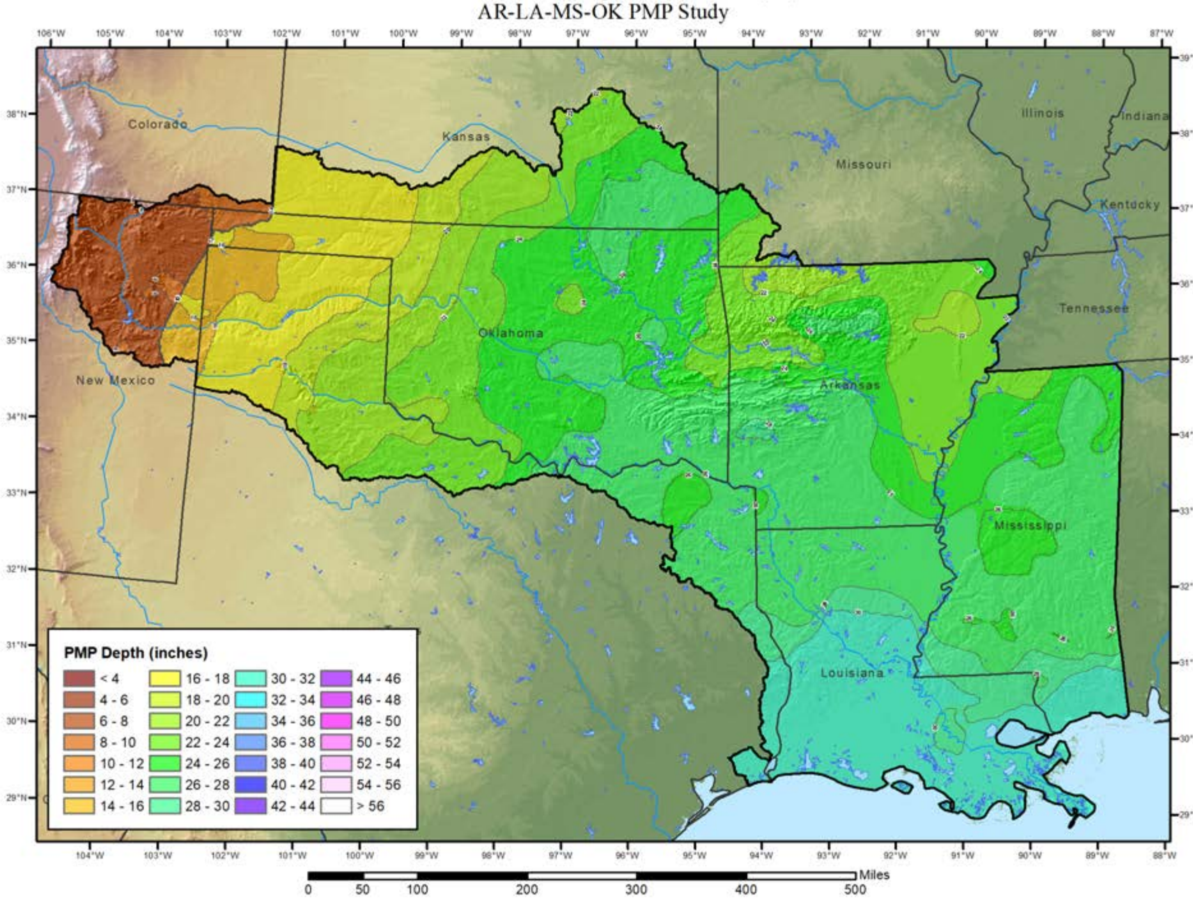
10 mi$^2$ 72-hour - Tropical Storm PMP (in.)
AR-LA-MS-OK PMP Study



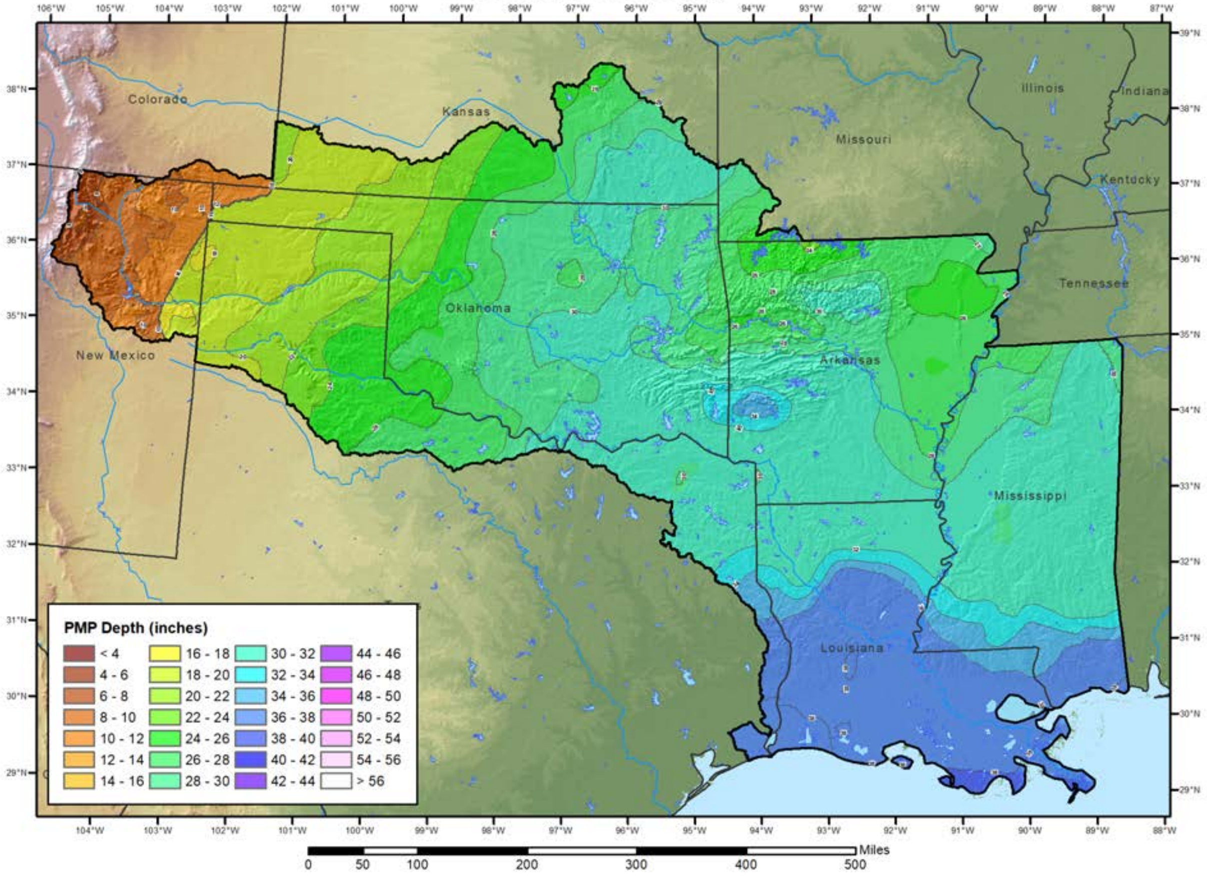
200 mi$^2$ 6-hour - Tropical Storm PMP (in.)
AR-LA-MS-OK PMP Study



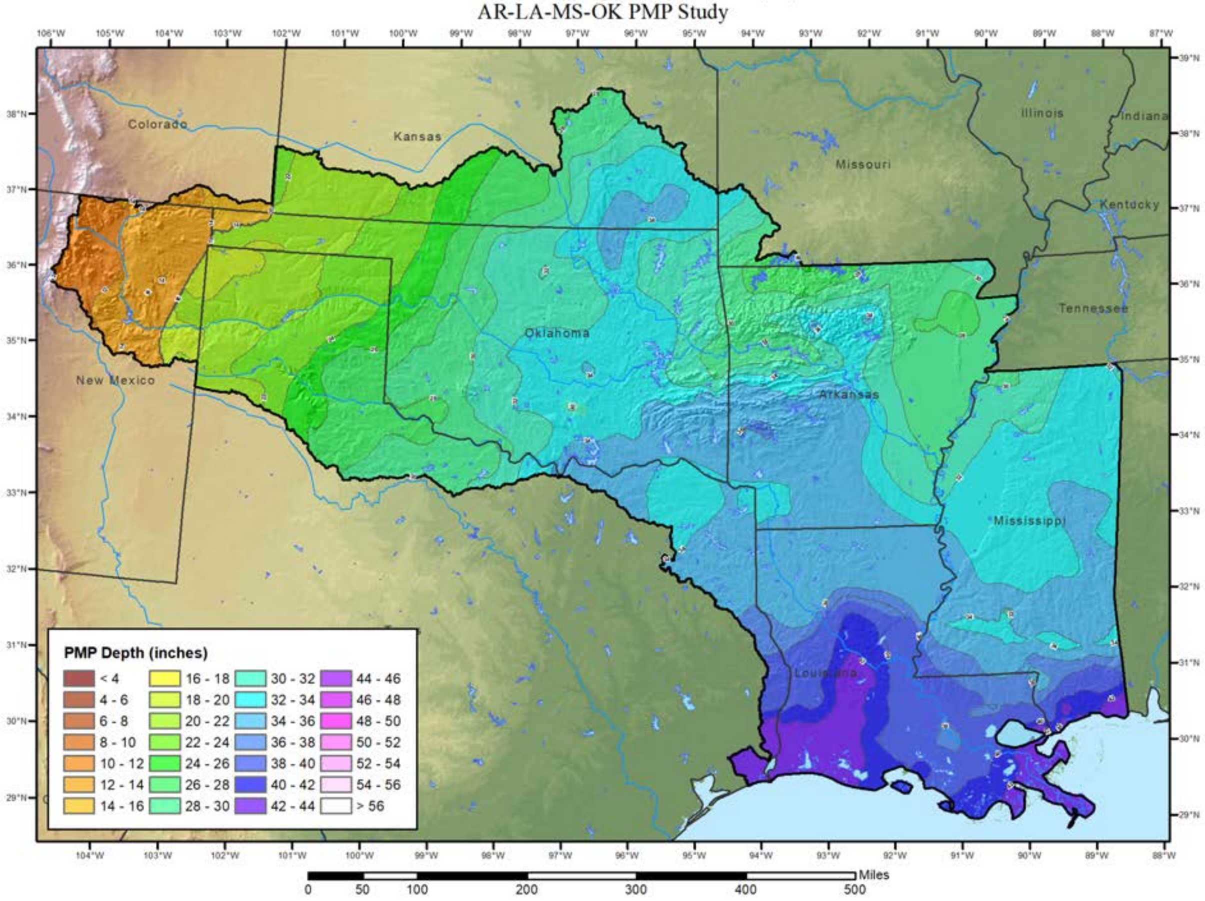
200 mi$^2$ 12-hour - Tropical Storm PMP (in.)
AR-LA-MS-OK PMP Study



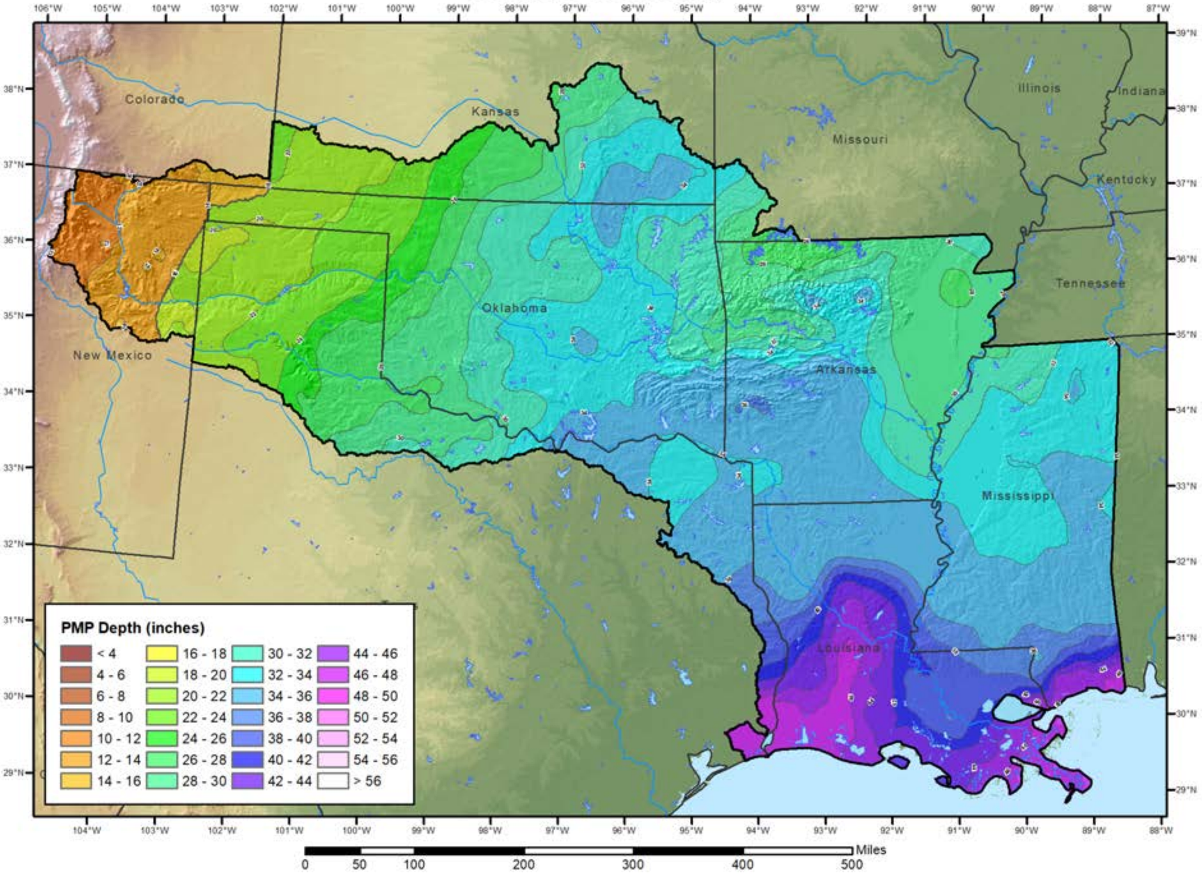
200 mi$^2$ 24-hour - Tropical Storm PMP (in.)
AR-LA-MS-OK PMP Study



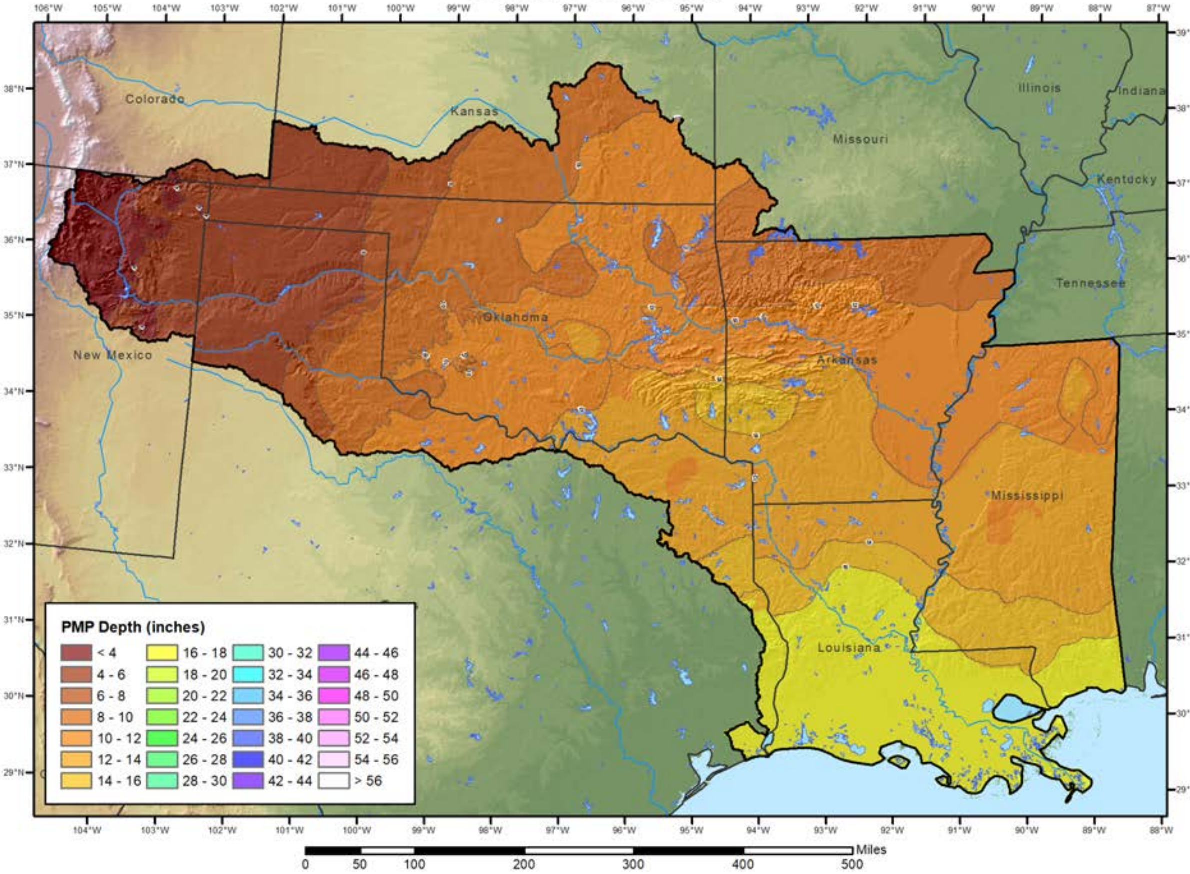
200 mi$^2$ 48-hour - Tropical Storm PMP (in.)
AR-LA-MS-OK PMP Study



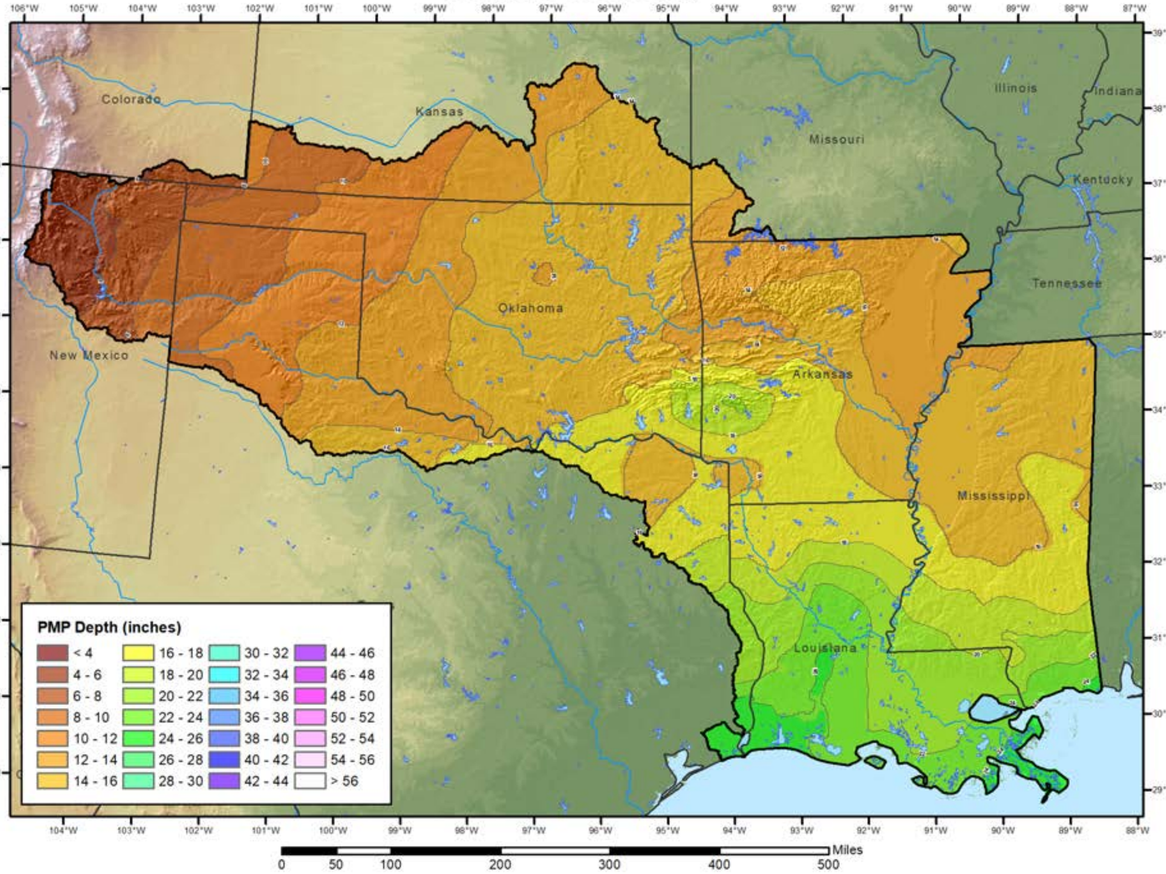
200 mi$^2$ 72-hour - Tropical Storm PMP (in.)
AR-LA-MS-OK PMP Study



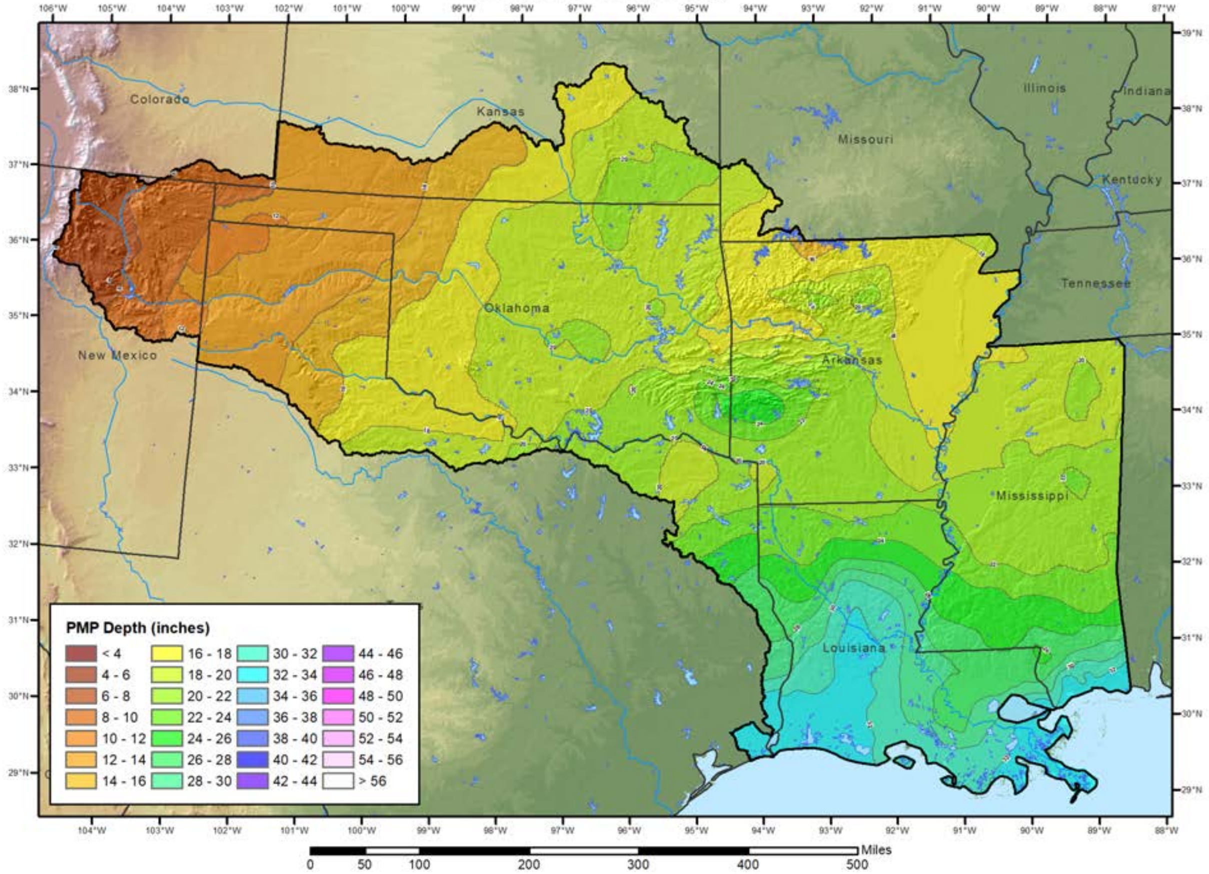
1,000 mi$^2$ 6-hour - Tropical Storm PMP (in.)
AR-LA-MS-OK PMP Study



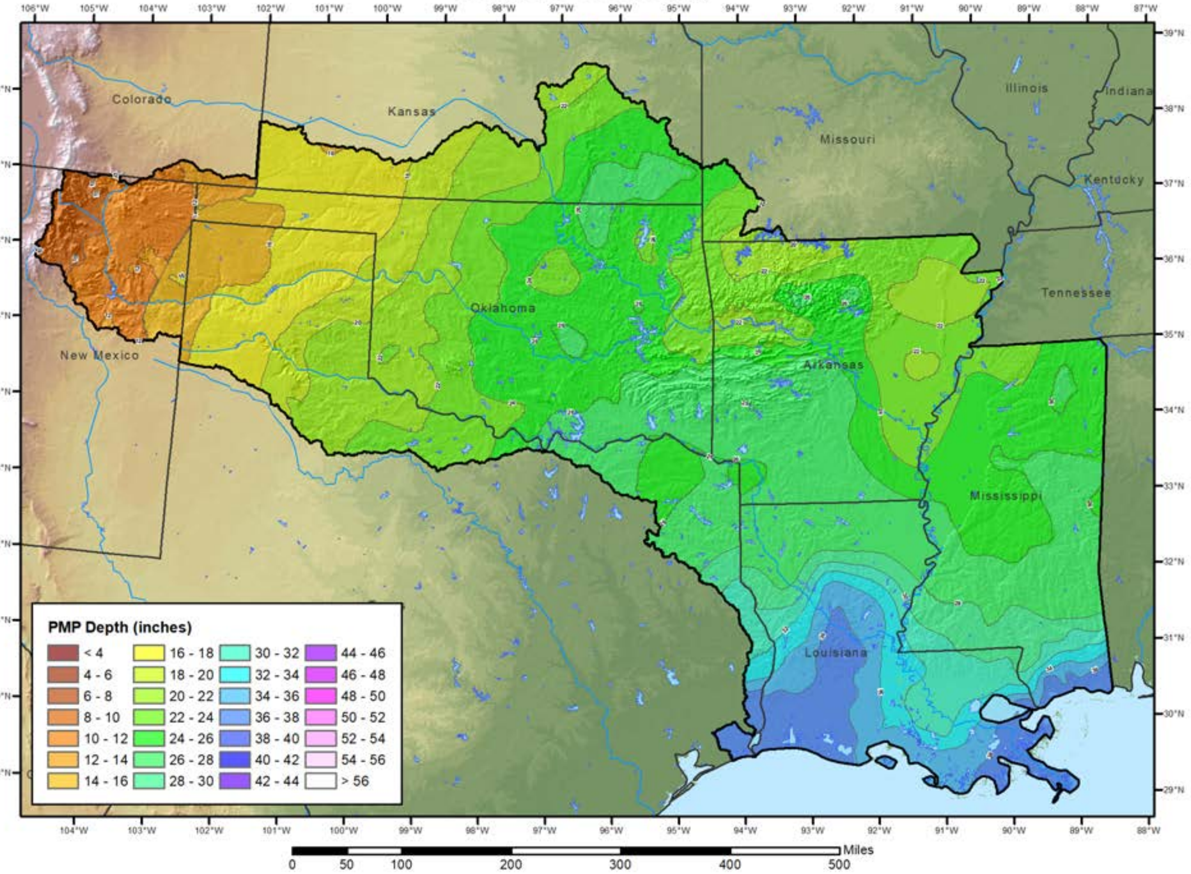
1,000 mi$^2$ 12-hour - Tropical Storm PMP (in.)
AR-LA-MS-OK PMP Study



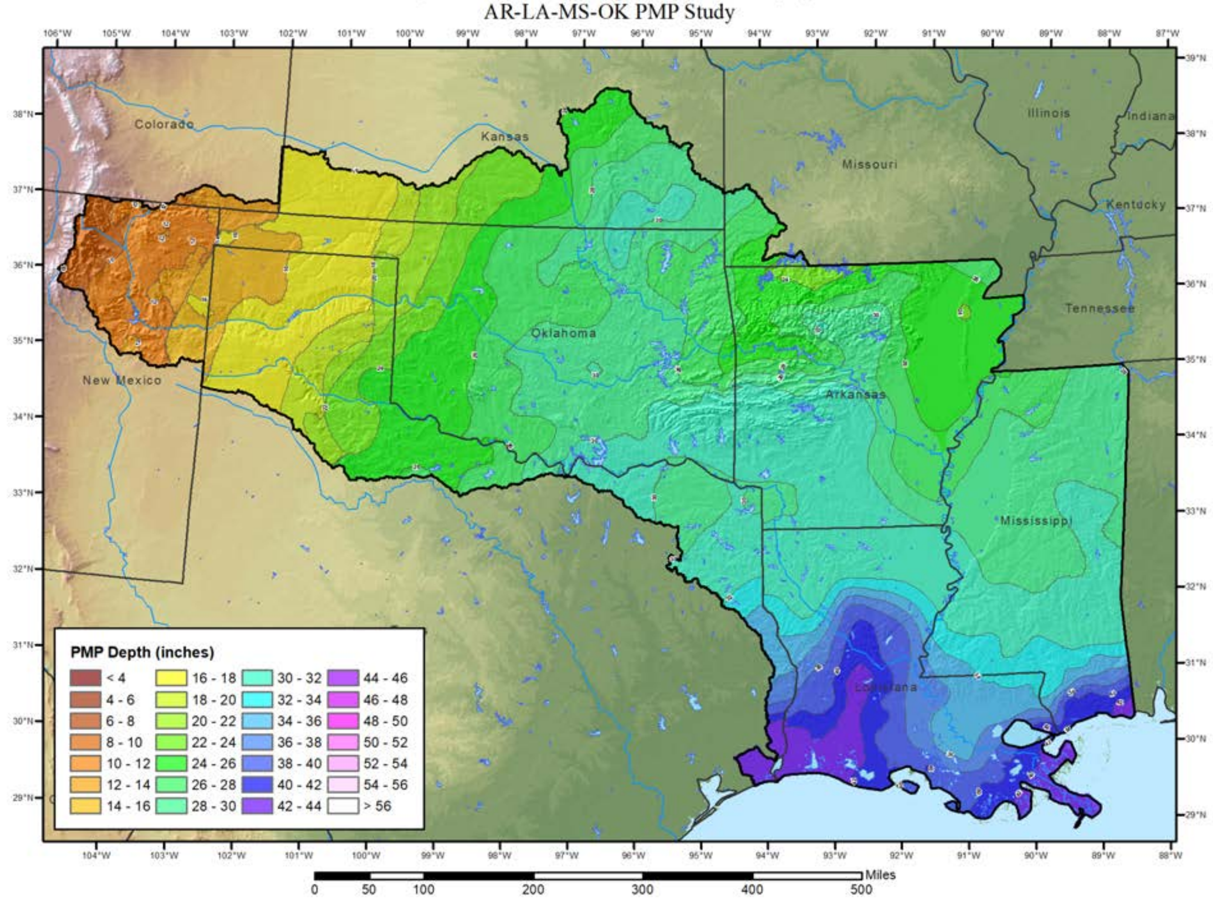
1,000 $mi^2$ 24-hour - Tropical Storm PMP (in.)
AR-LA-MS-OK PMP Study



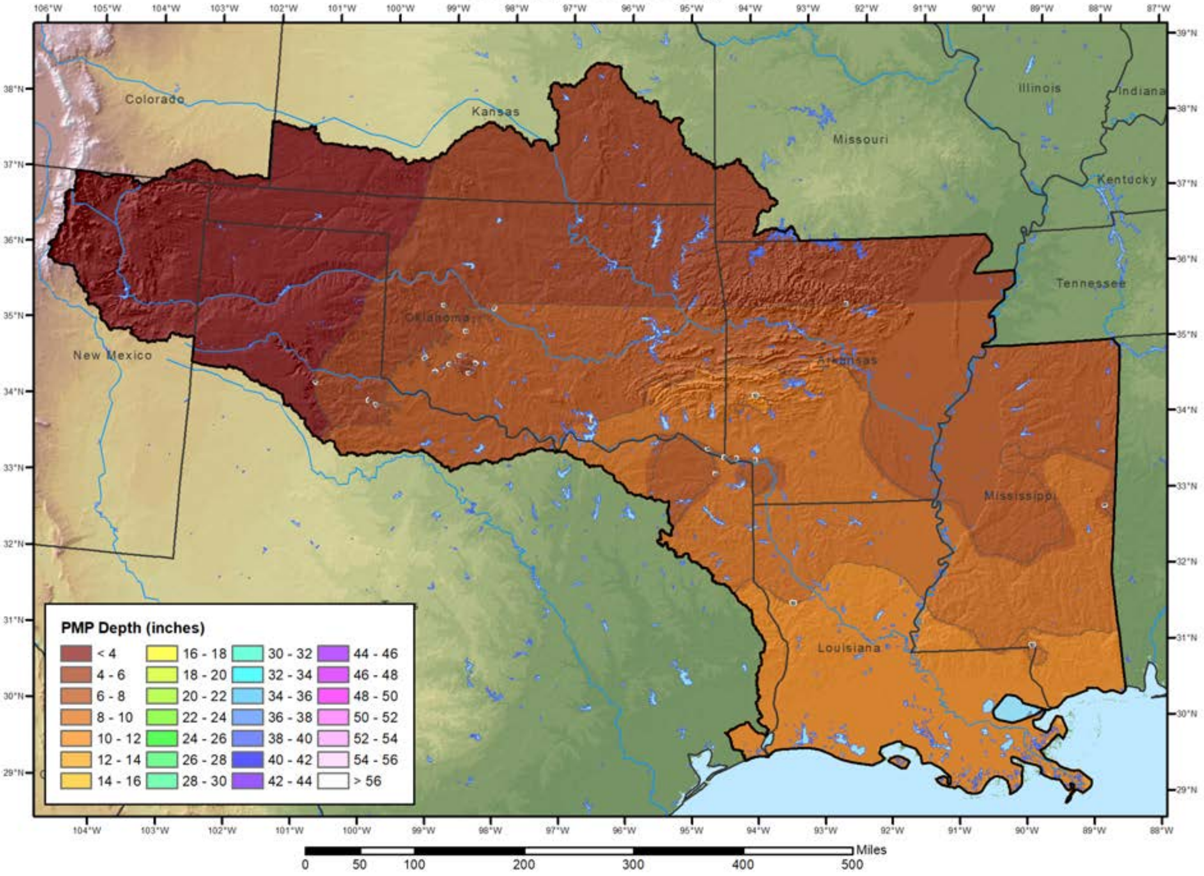
1,000 mi$^2$ 48-hour - Tropical Storm PMP (in.)
AR-LA-MS-OK PMP Study



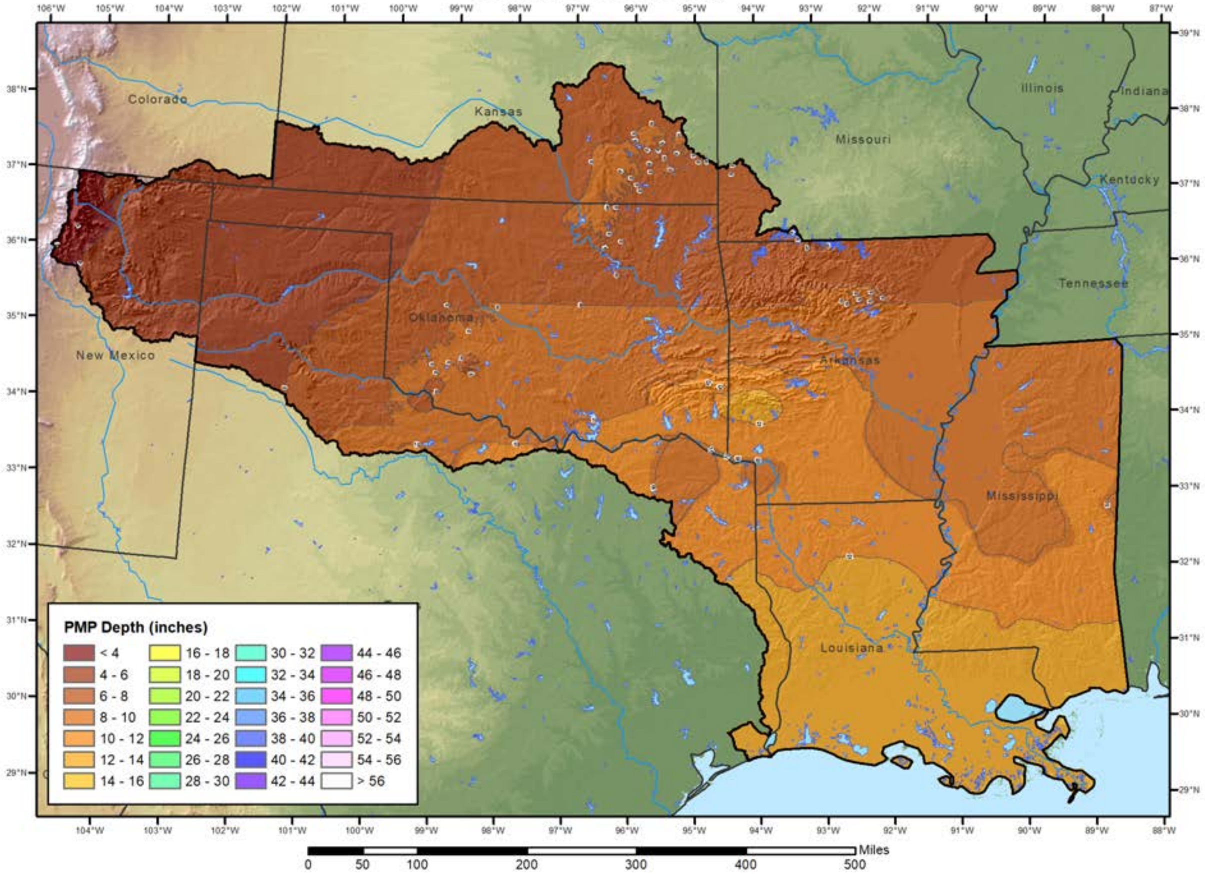
1,000 mi$^2$ 72-hour - Tropical Storm PMP (in.)
AR-LA-MS-OK PMP Study



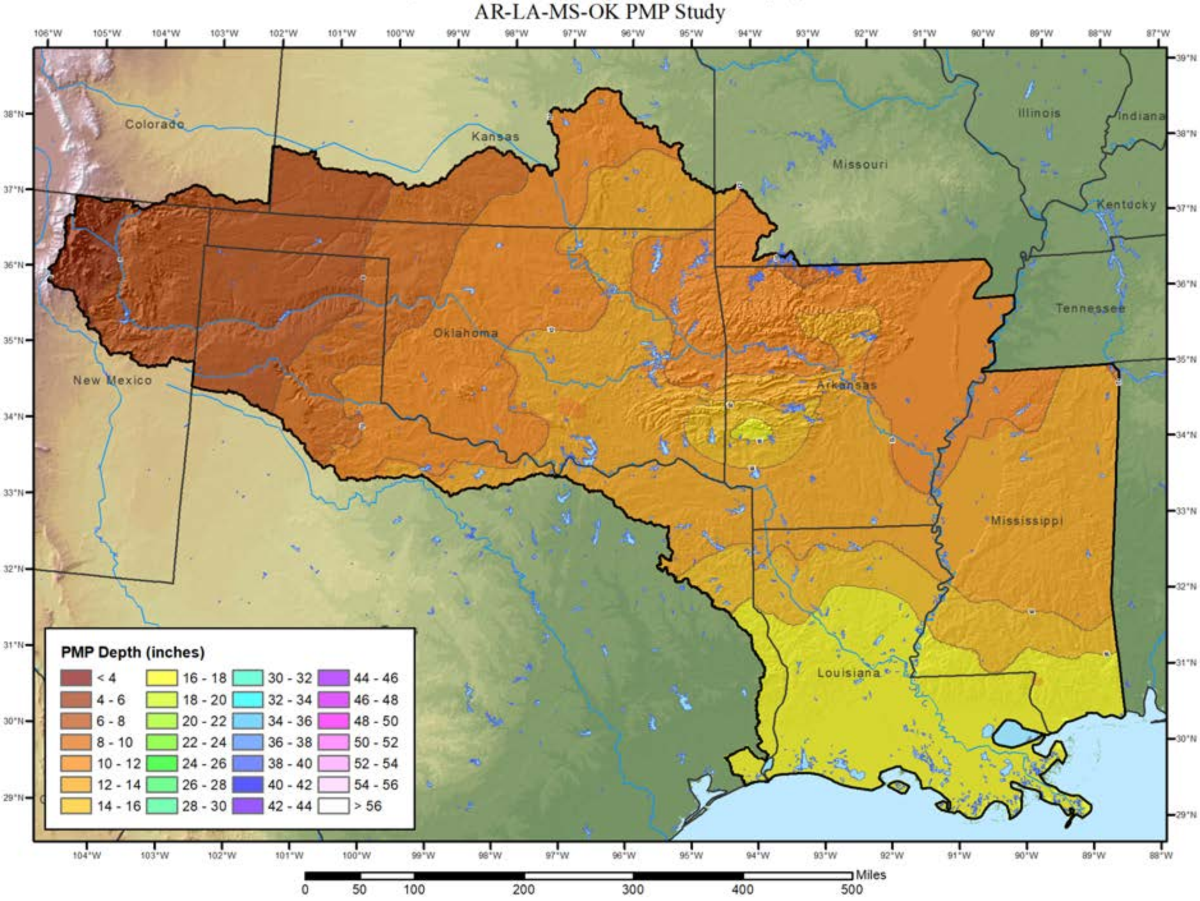
5,000 mi$^2$ 6-hour - Tropical Storm PMP (in.)
AR-LA-MS-OK PMP Study



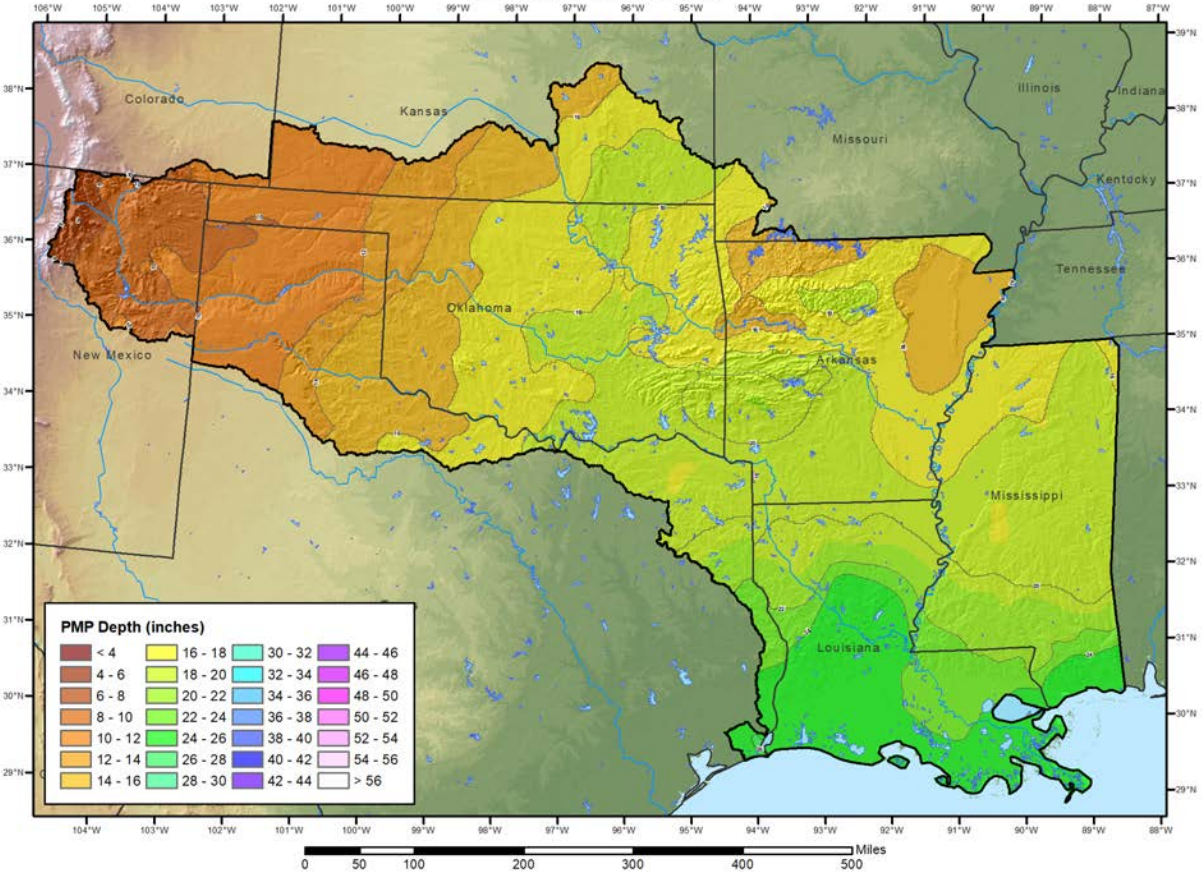
5,000 mi$^2$ 12-hour - Tropical Storm PMP (in.)
AR-LA-MS-OK PMP Study



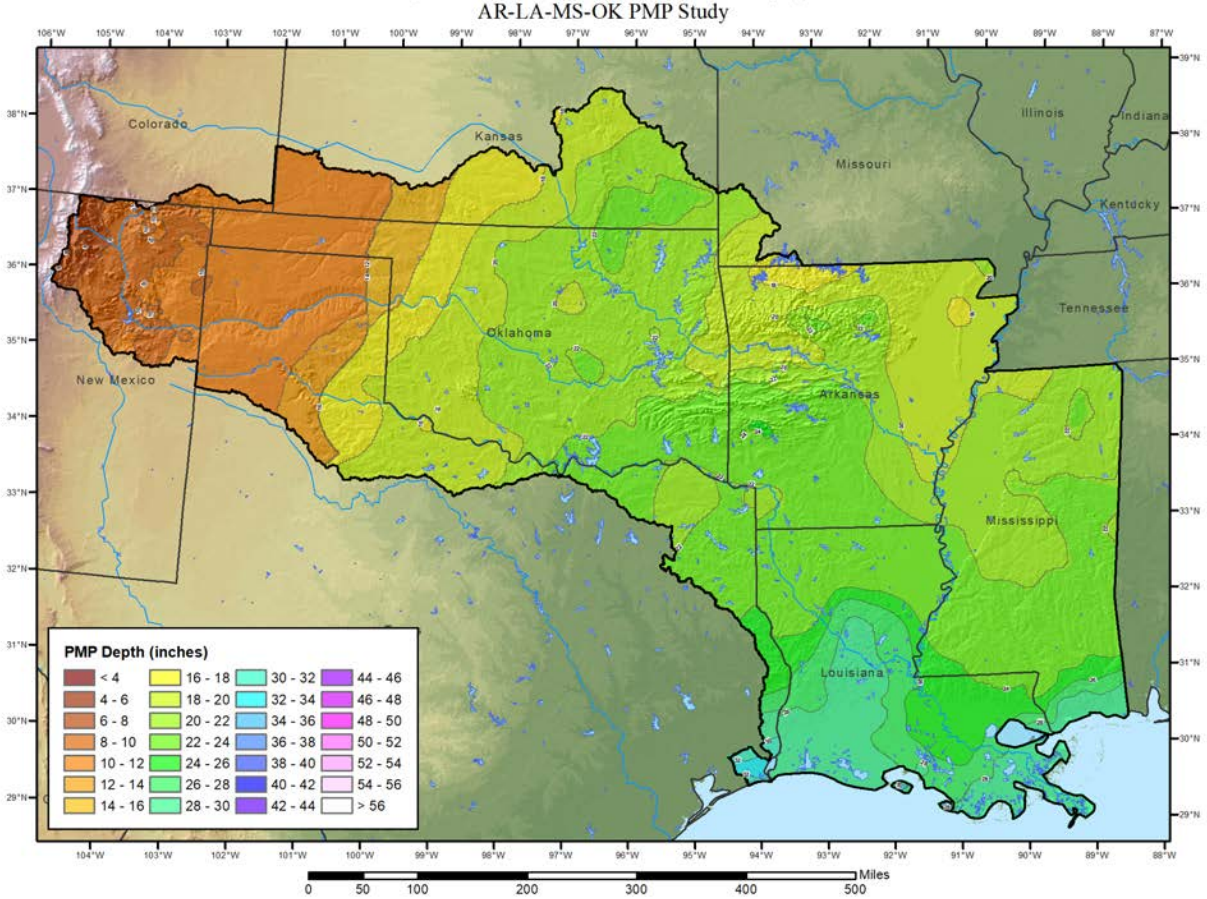
5,000 mi$^2$ 24-hour - Tropical Storm PMP (in.)
AR-LA-MS-OK PMP Study



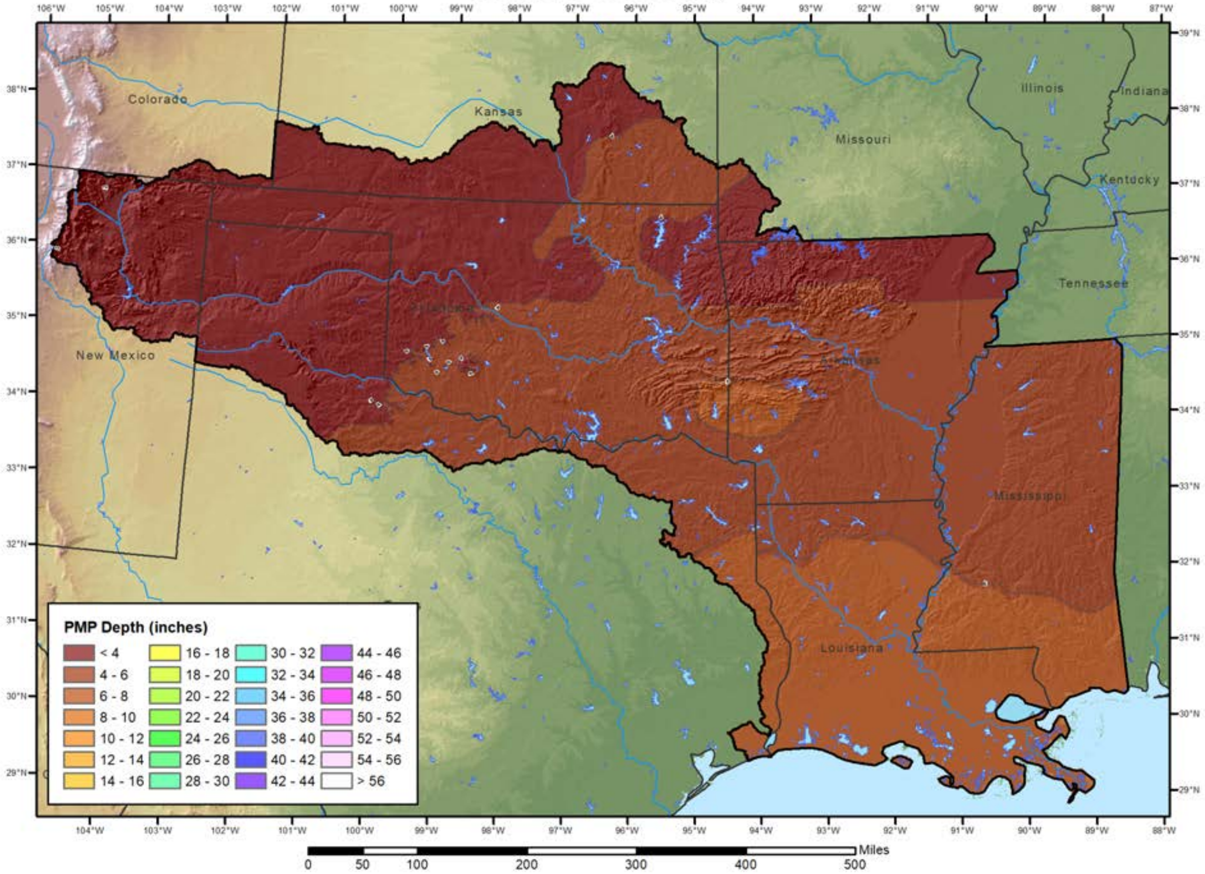
5,000 mi$^2$ 48-hour - Tropical Storm PMP (in.)
AR-LA-MS-OK PMP Study



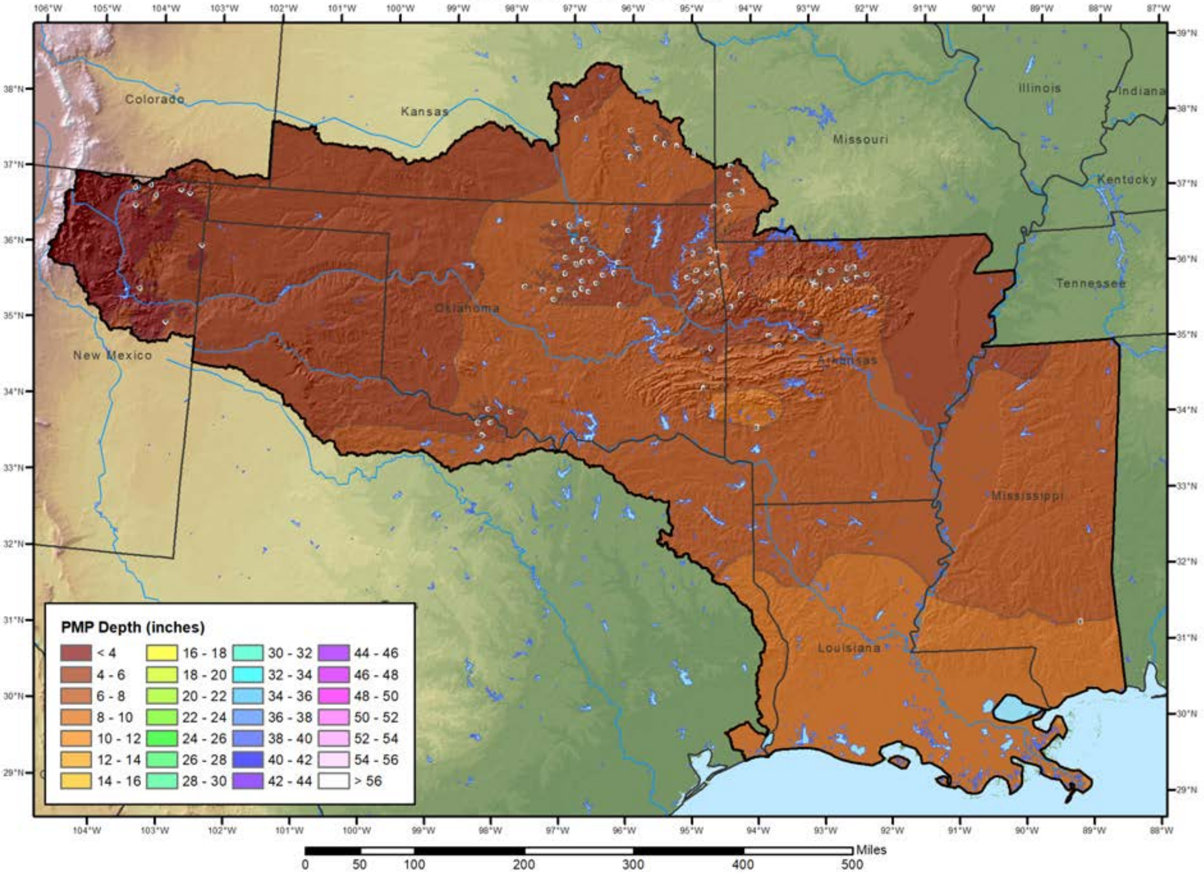
5,000 mi$^2$ 72-hour - Tropical Storm PMP (in.)
AR-LA-MS-OK PMP Study



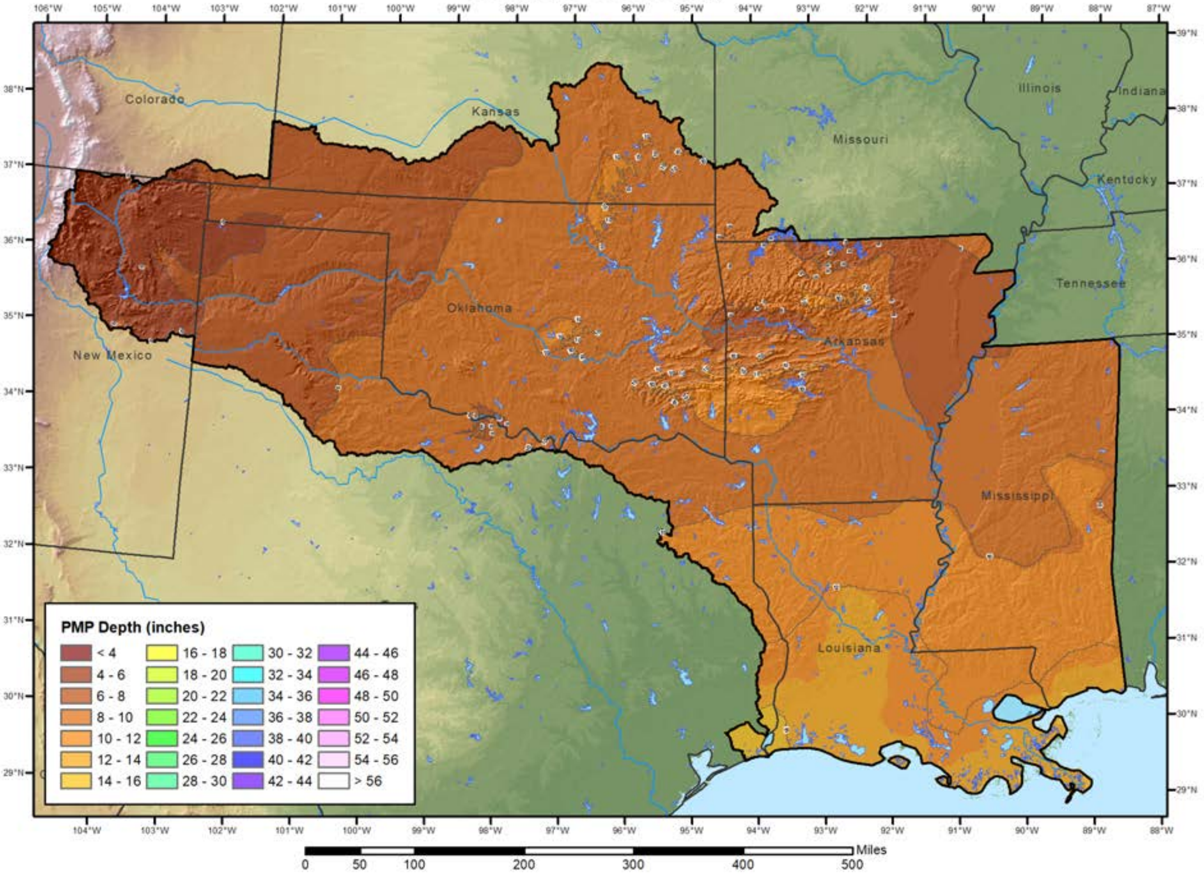
10,000 mi$^2$ 6-hour - Tropical Storm PMP (in.)
AR-LA-MS-OK PMP Study



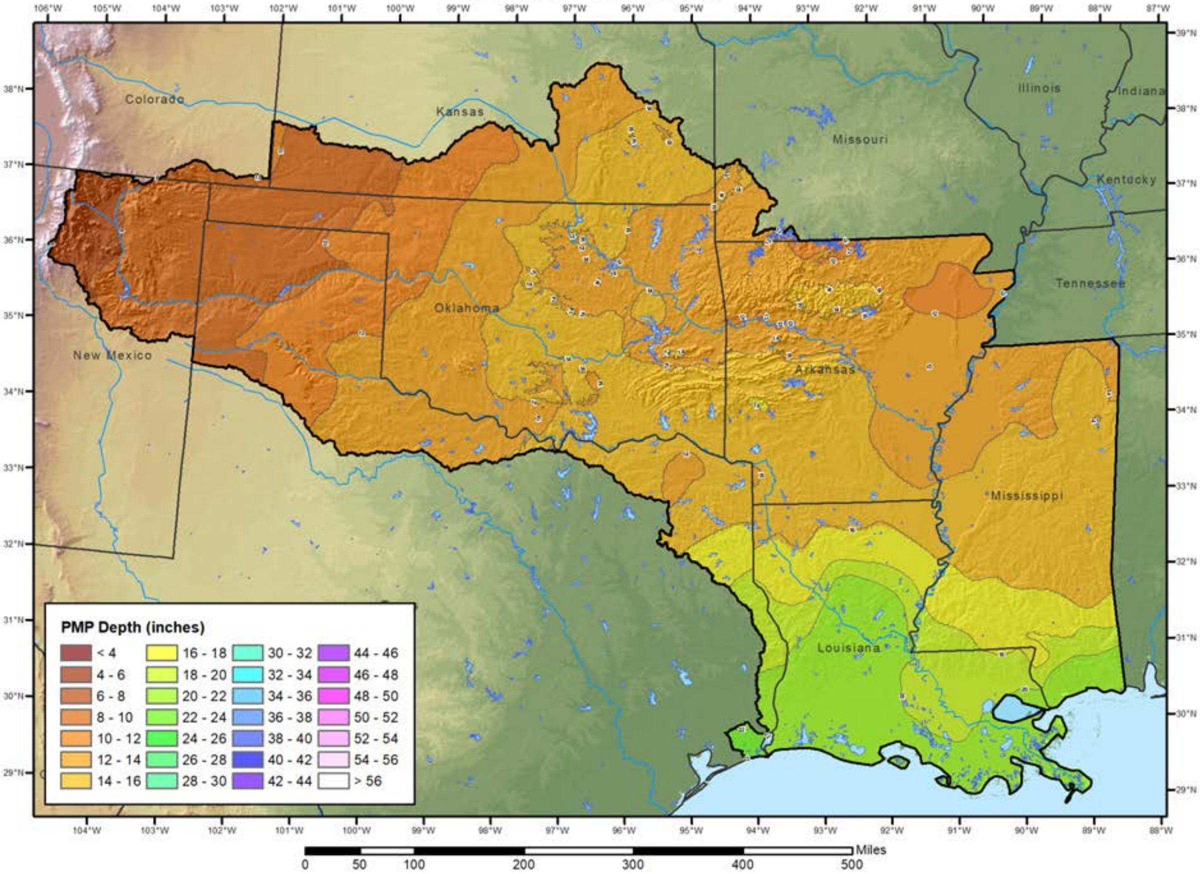
10,000 mi$^2$ 12-hour - Tropical Storm PMP (in.)
AR-LA-MS-OK PMP Study



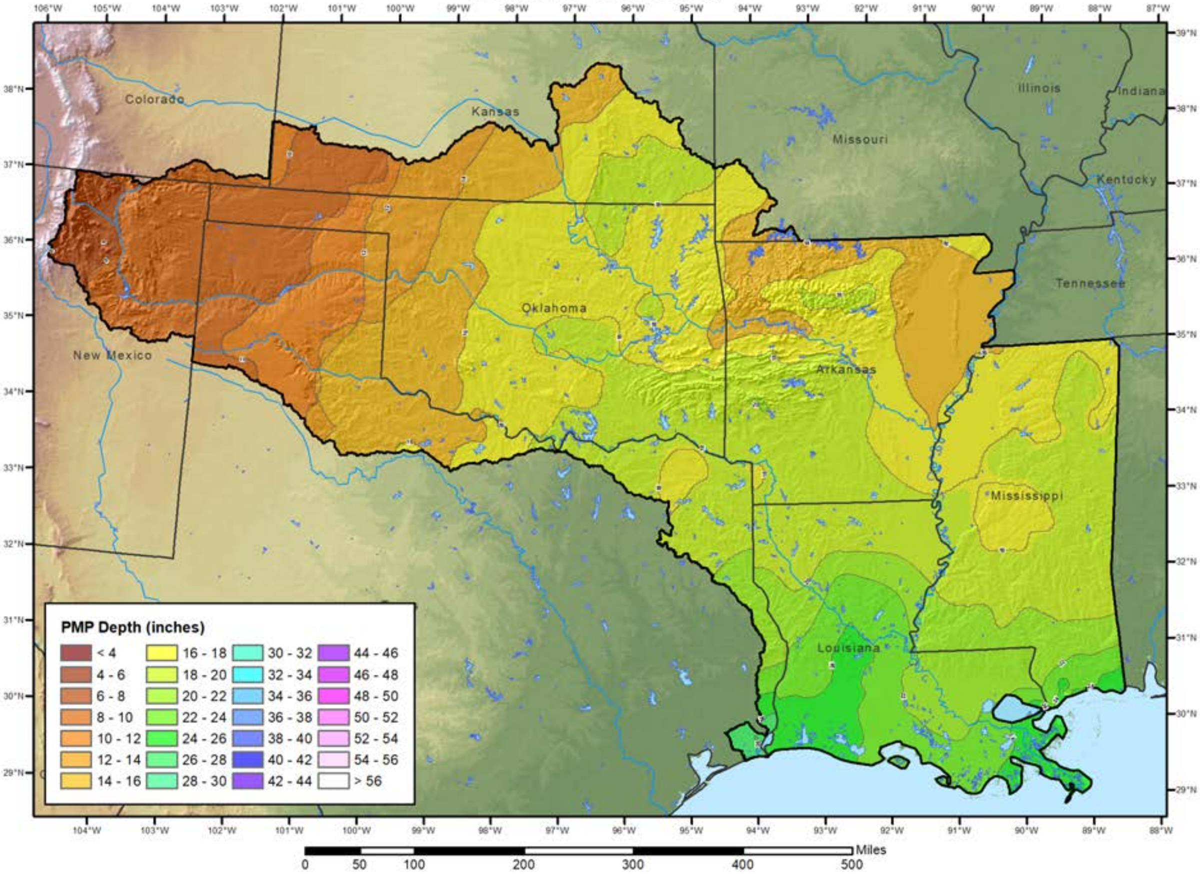
10,000 mi$^2$ 24-hour - Tropical Storm PMP (in.)
AR-LA-MS-OK PMP Study



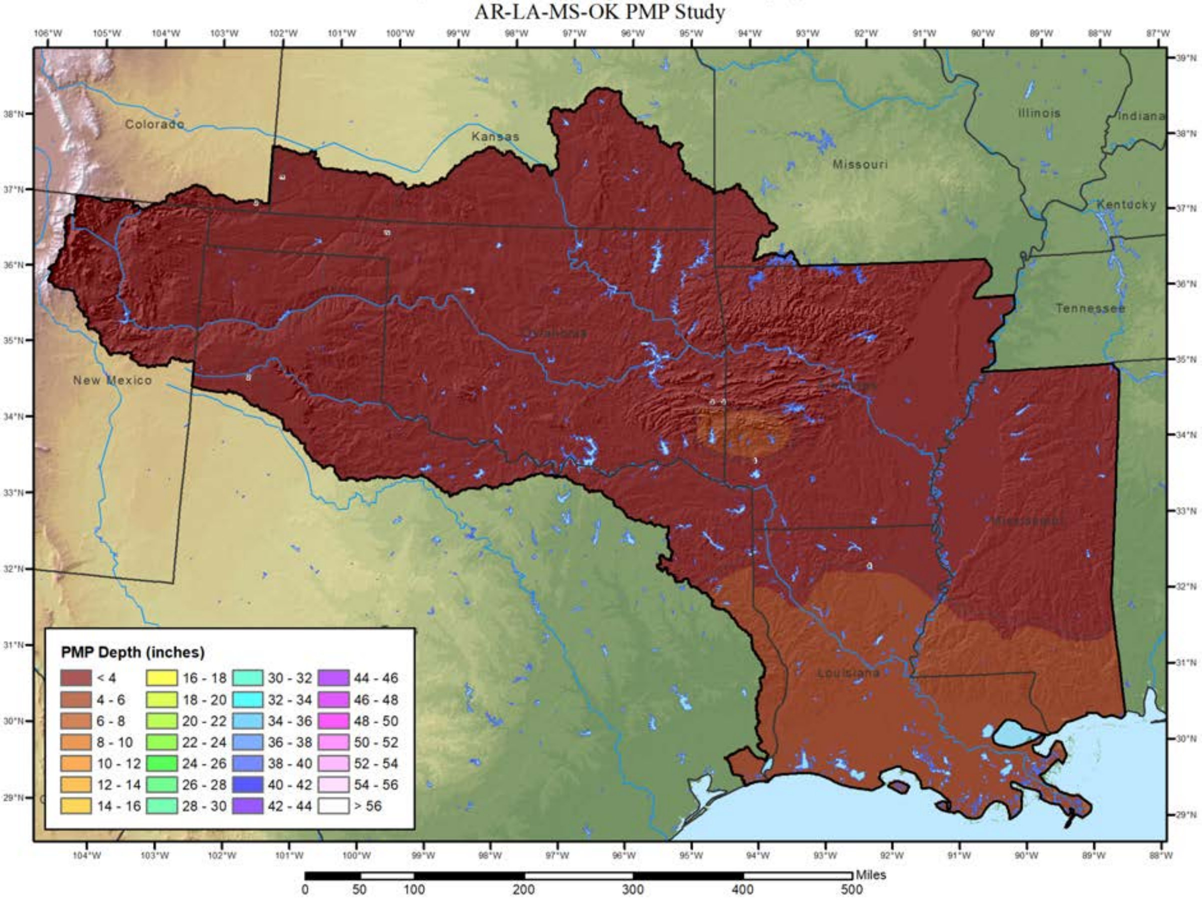
10,000 mi$^2$ 48-hour - Tropical Storm PMP (in.)
AR-LA-MS-OK PMP Study



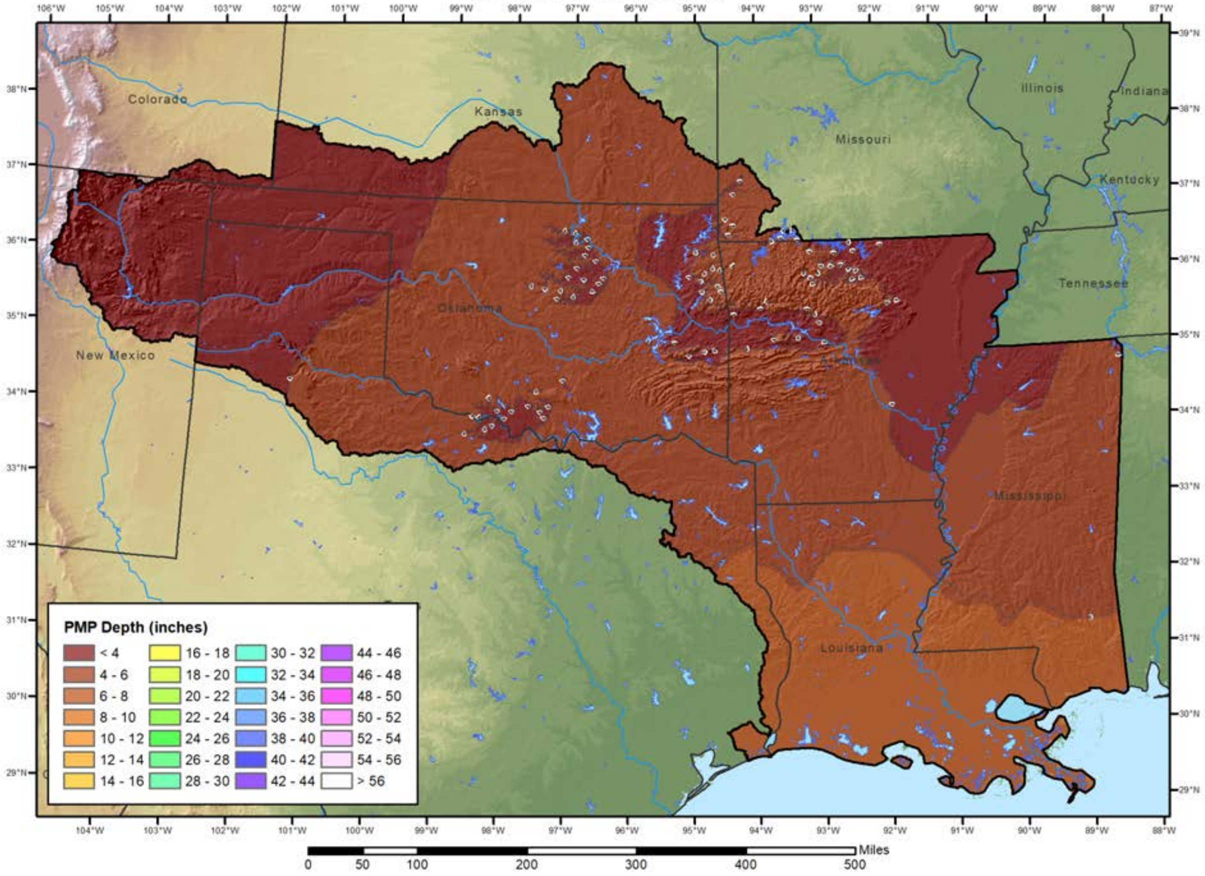
10,000 mi$^2$ 72-hour - Tropical Storm PMP (in.)
AR-LA-MS-OK PMP Study



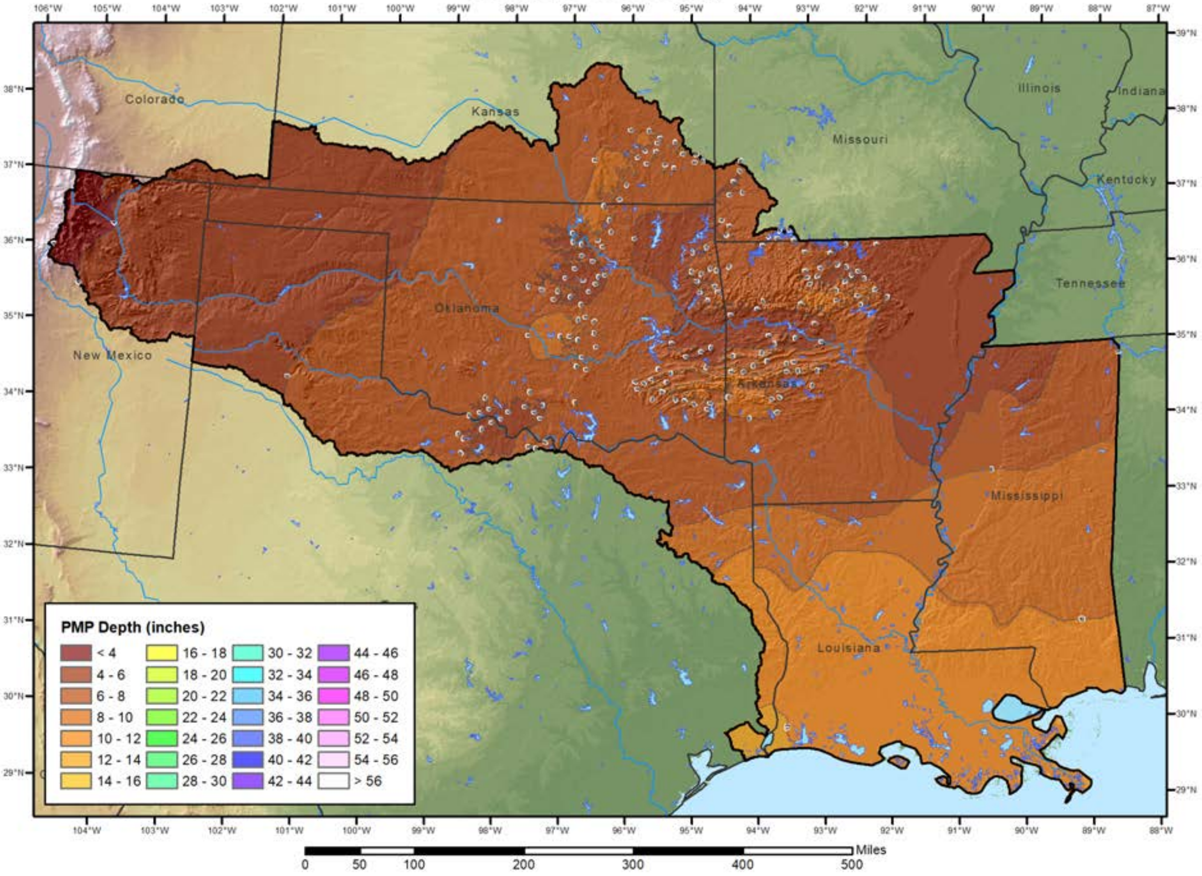
20,000 mi$^2$ 6-hour - Tropical Storm PMP (in.)
AR-LA-MS-OK PMP Study



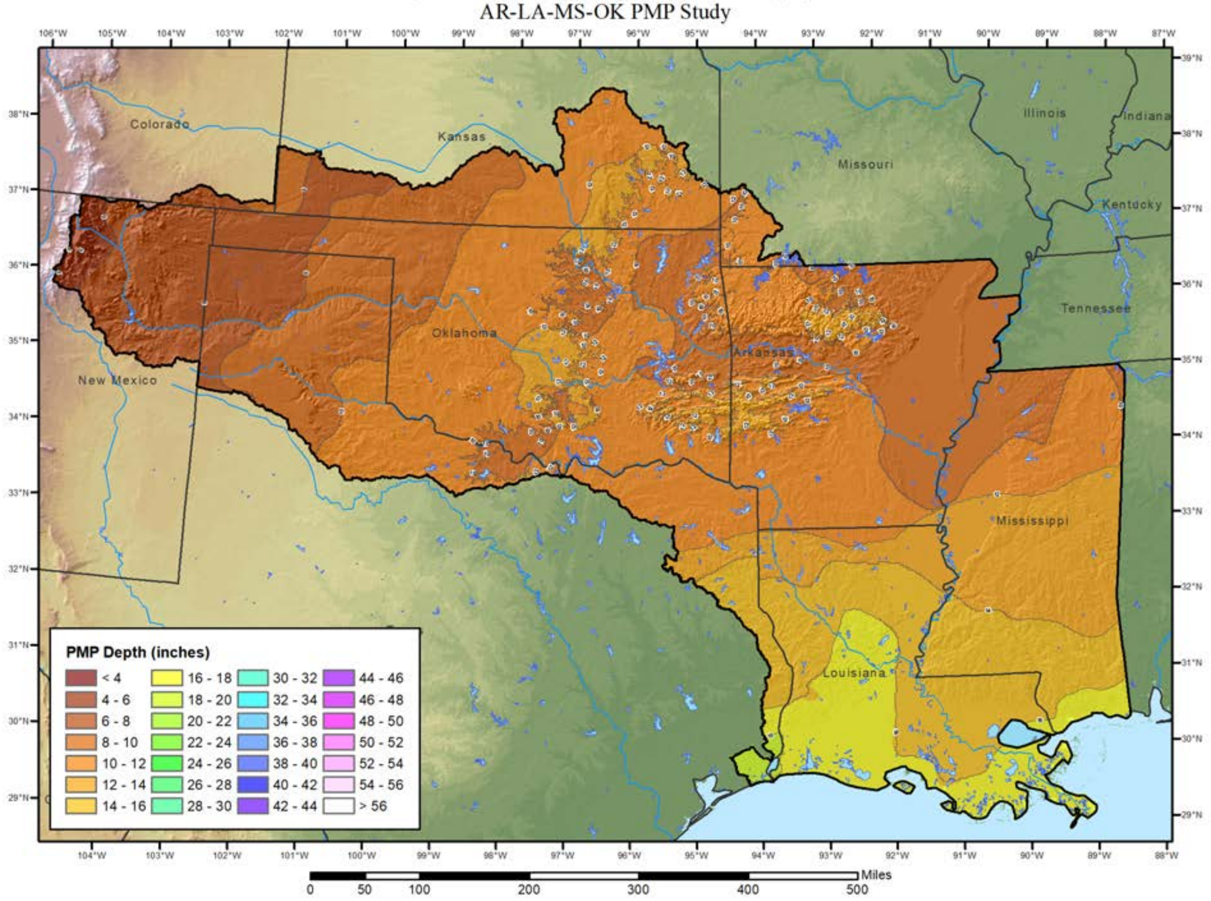
20,000 mi$^2$ 12-hour - Tropical Storm PMP (in.)
AR-LA-MS-OK PMP Study



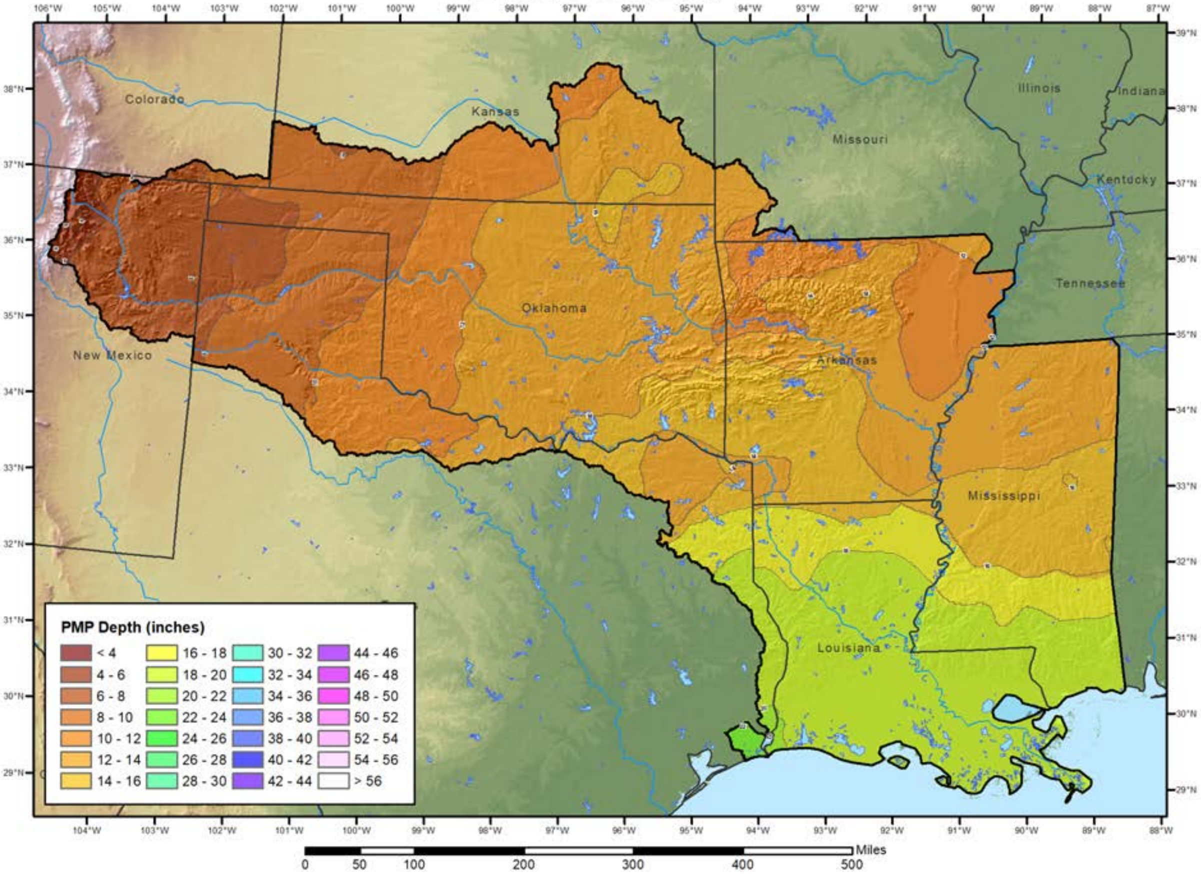
20,000 mi$^2$ 24-hour - Tropical Storm PMP (in.)
AR-LA-MS-OK PMP Study



20,000 mi$^2$ 48-hour - Tropical Storm PMP (in.)
AR-LA-MS-OK PMP Study



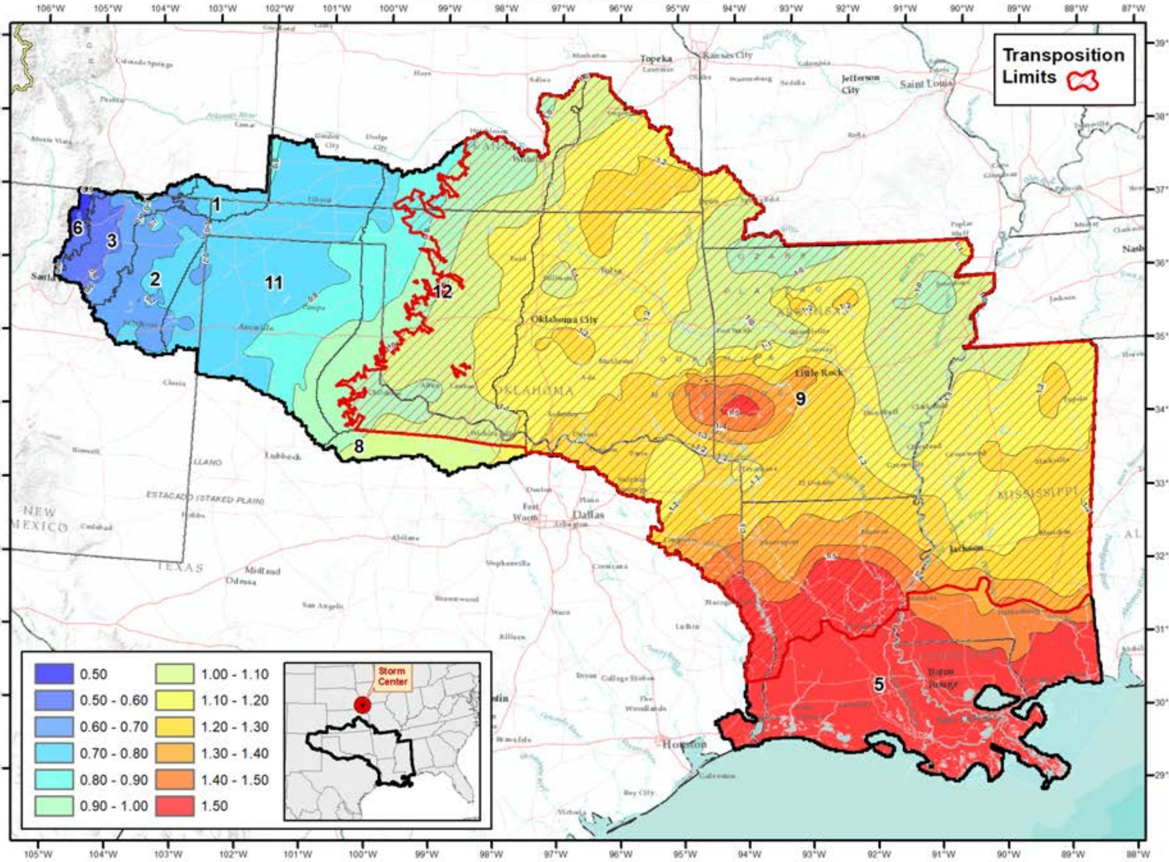
20,000 mi$^2$ 72-hour - Tropical Storm PMP (in.)
AR-LA-MS-OK PMP Study



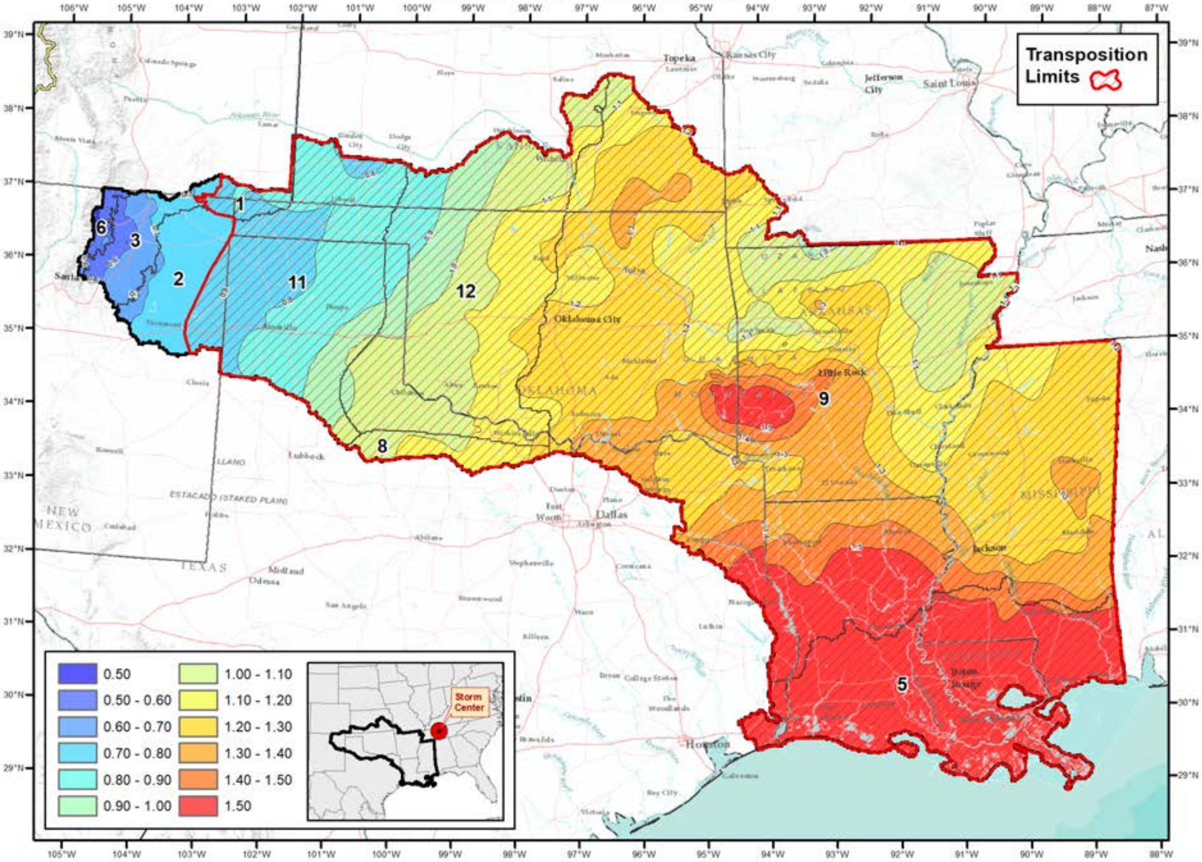
# Appendix B
## Geographic Transposition Factor (GTF) Maps

# General Storms

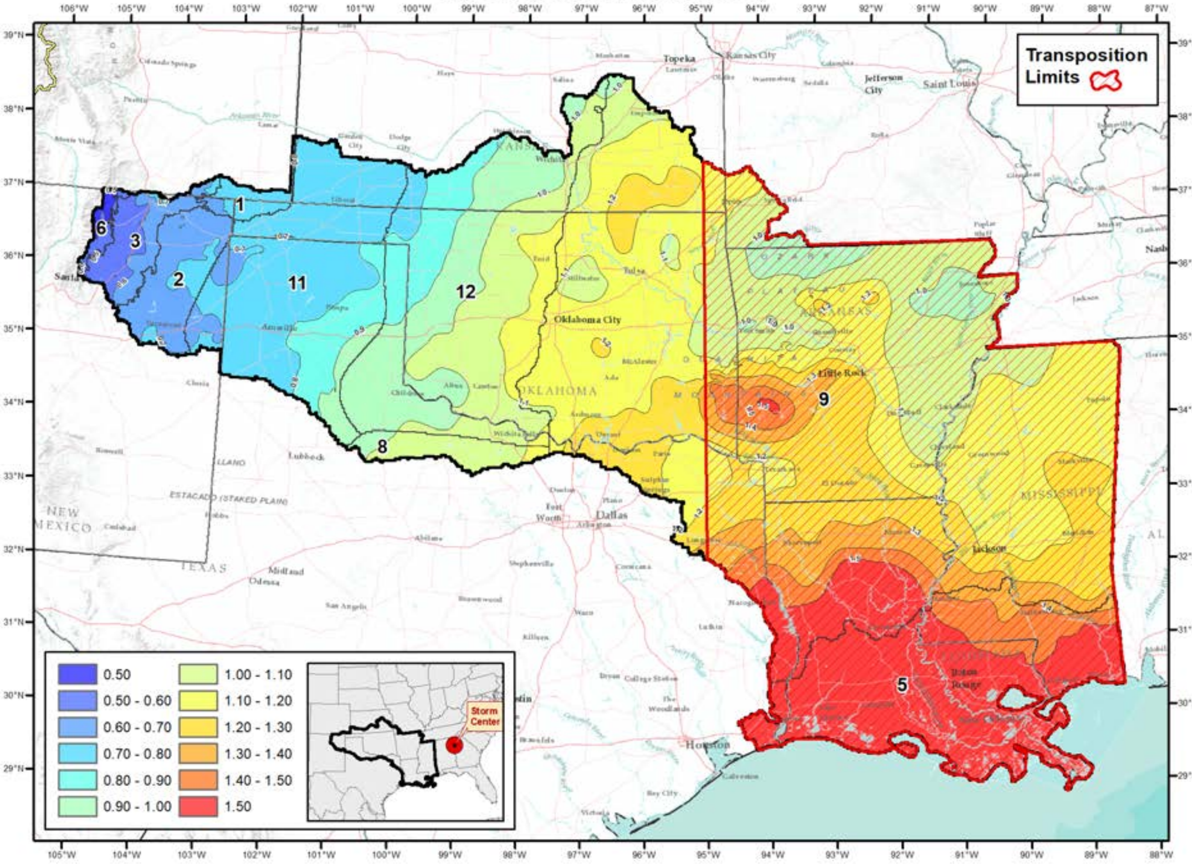
Gridded GTF for Edgerton, MO
Jul, 1965 (SPAS 1183 DAD zone 1)



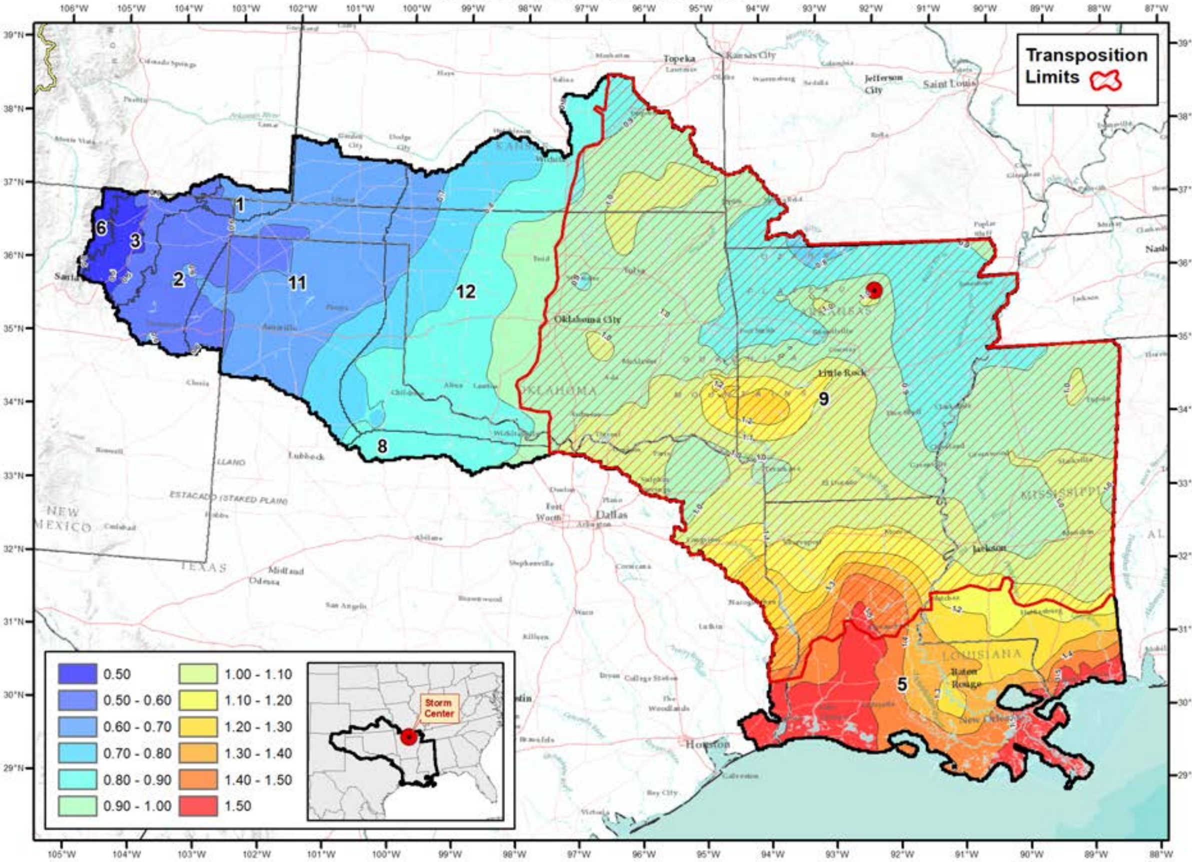
Gridded GTF for Warner Park, TN
May, 2010 (SPAS 1208 DAD zone 1)



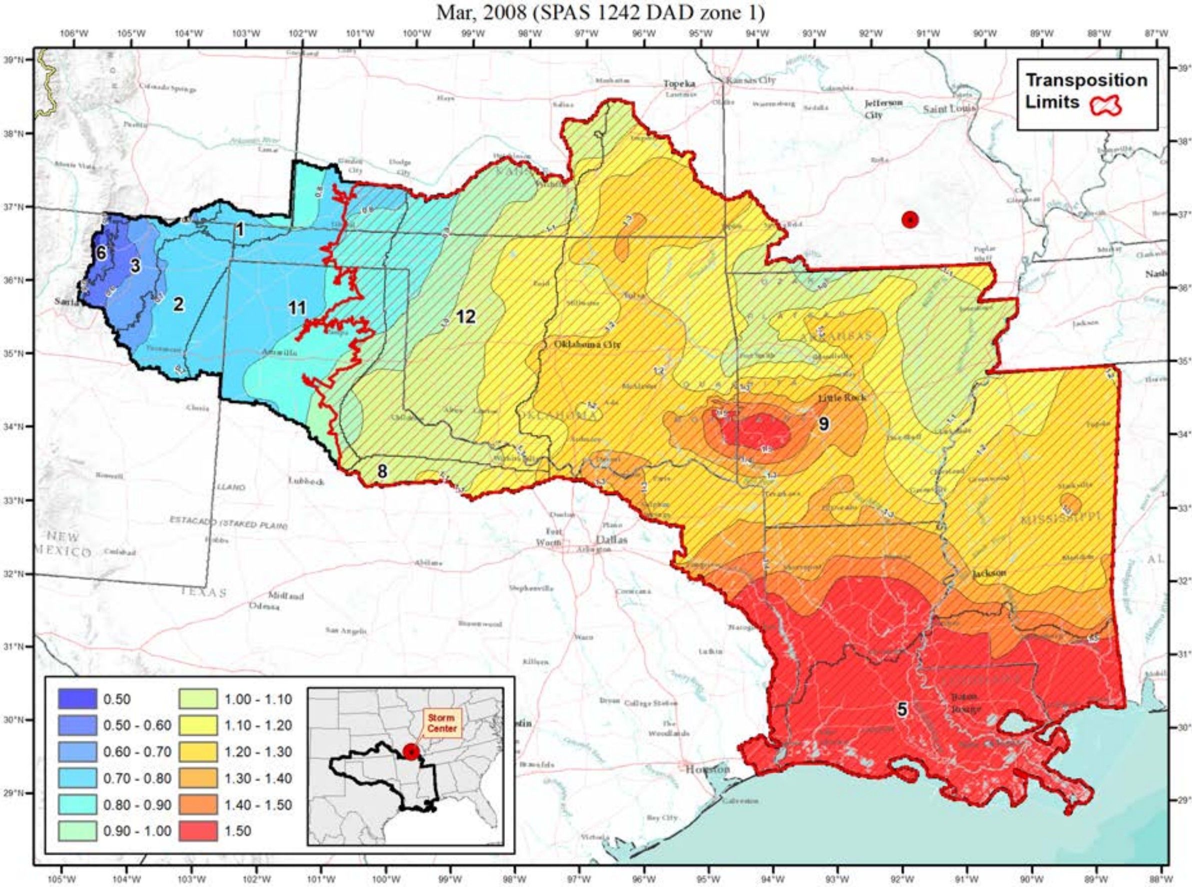
Gridded GTF for Douglasville, GA
Sep, 2009 (SPAS 1218 DAD zone 1)



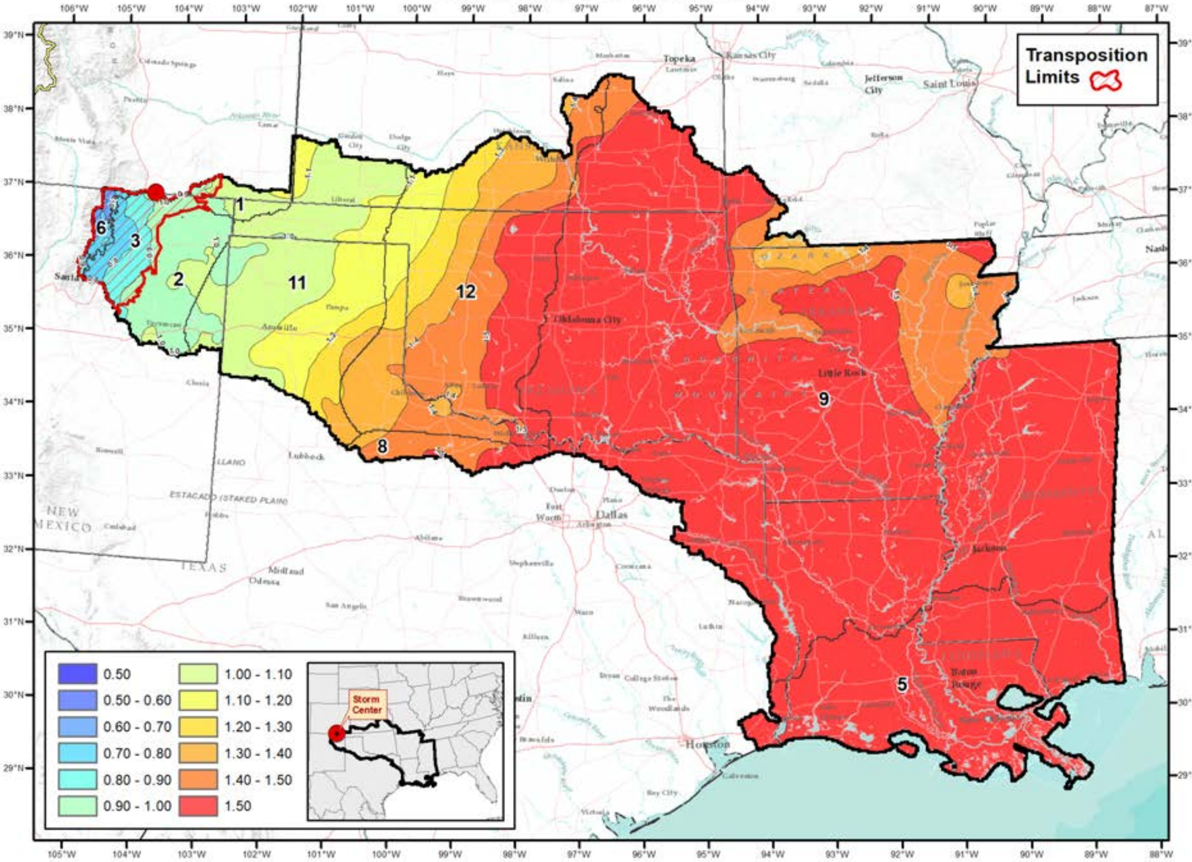
Gridded GTF for Big Fork, AR
Dec, 1982 (SPAS 1219 DAD zone 1)



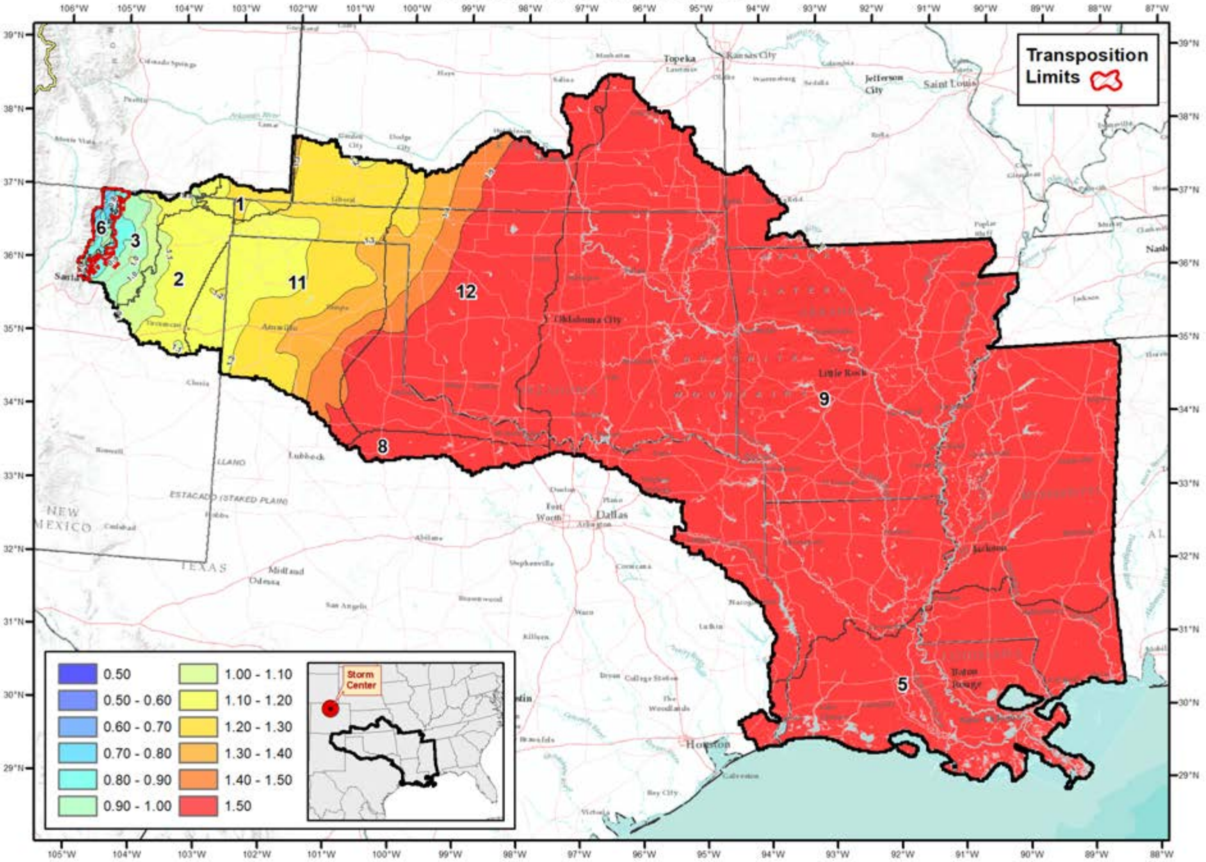
Gridded GTF for Alley Spring, MO
Mar, 2008 (SPAS 1242 DAD zone 1)



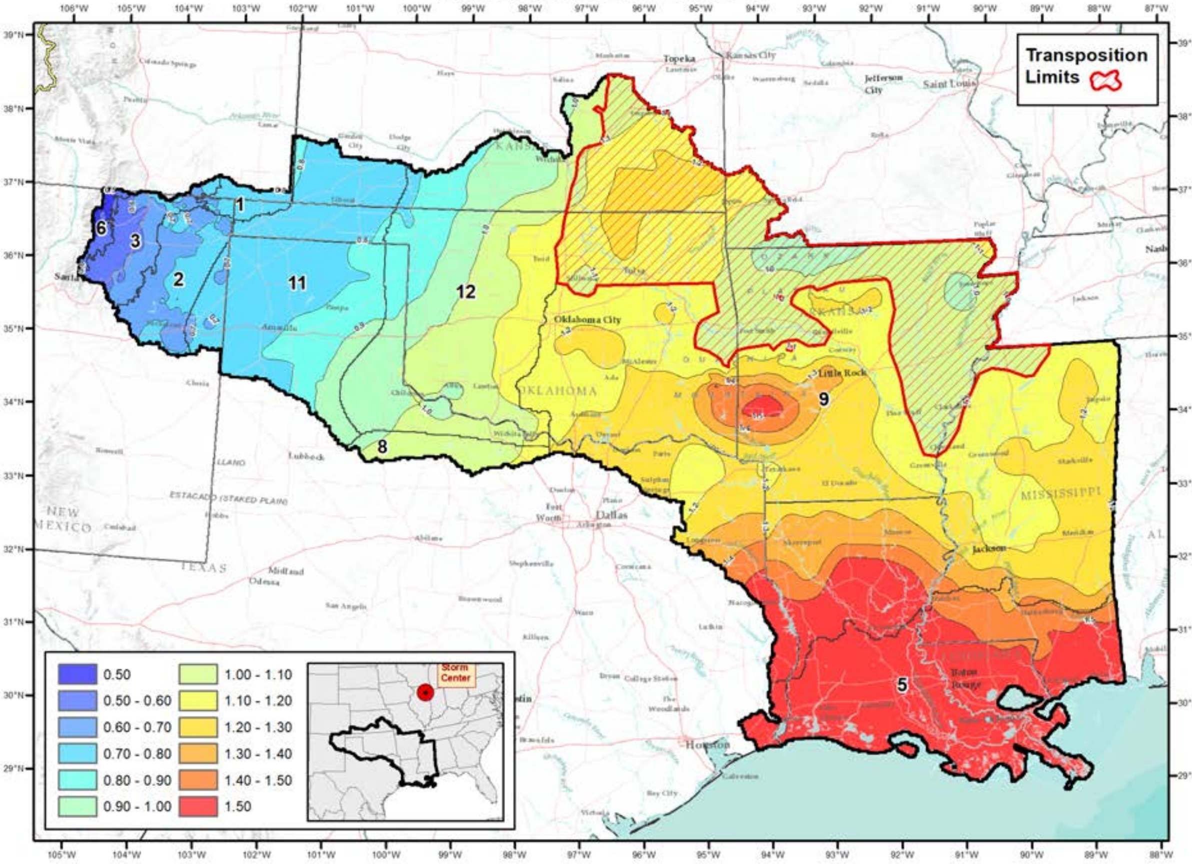
Gridded GTF for Lake Maloya, NM
May, 1955 (SPAS 1251 DAD zone 1)



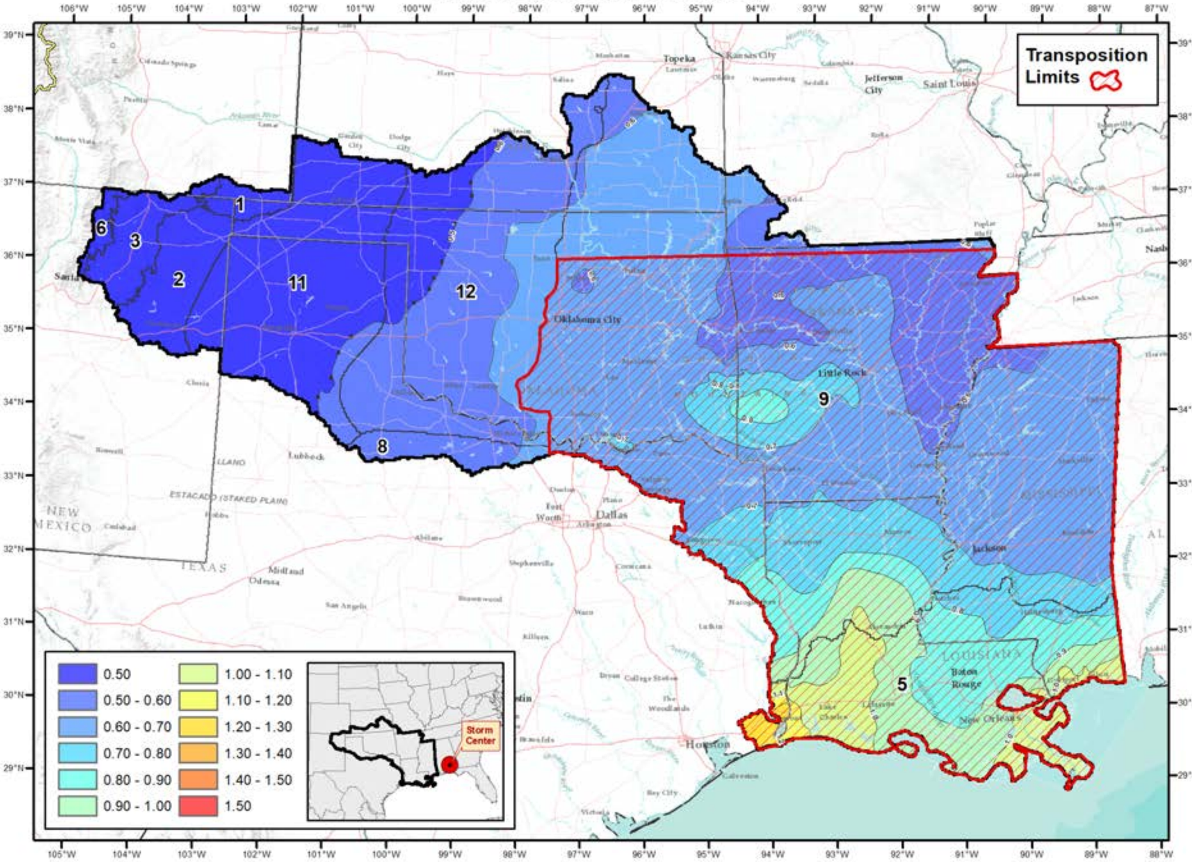
Gridded GTF for Big Elk Meadow, CO
May, 1969 (SPAS 1253 DAD zone 1)



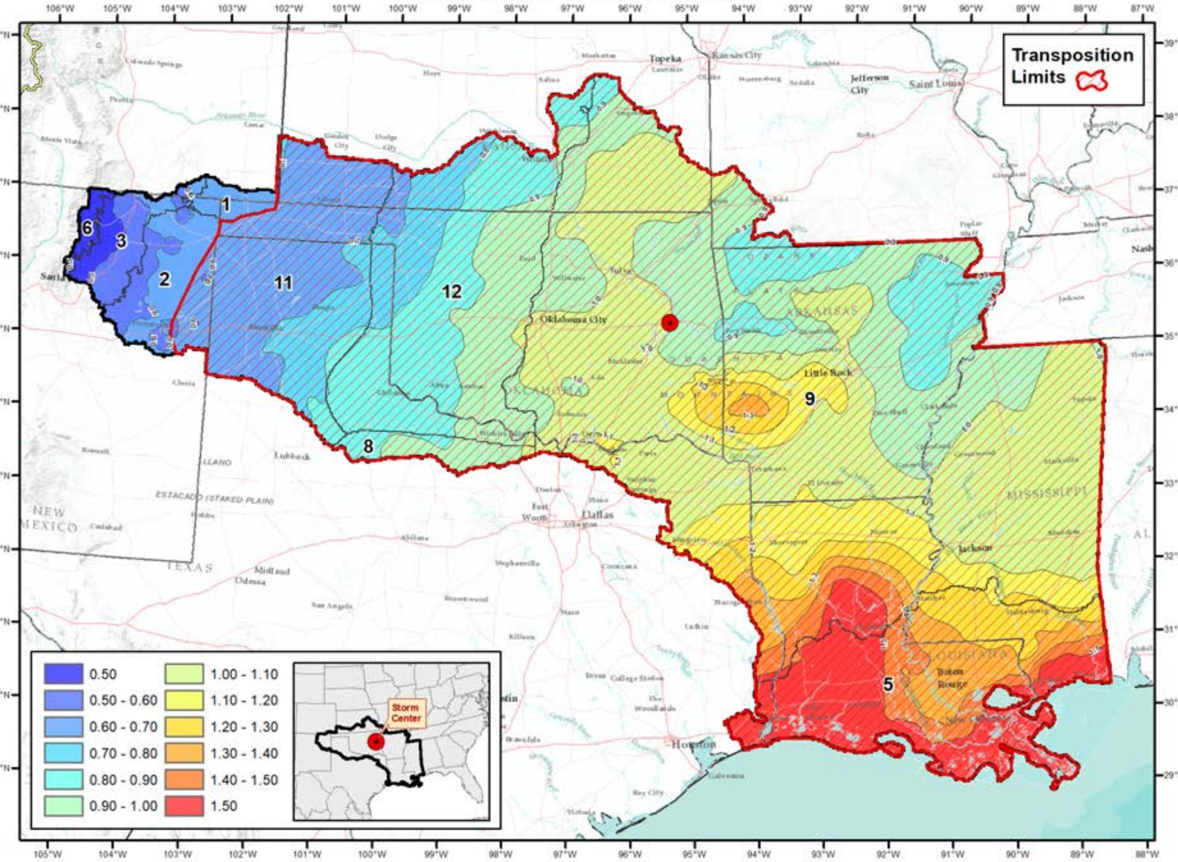
Gridded GTF for Aurora College, IL
Jul, 1996 (SPAS 1286 DAD zone 1)



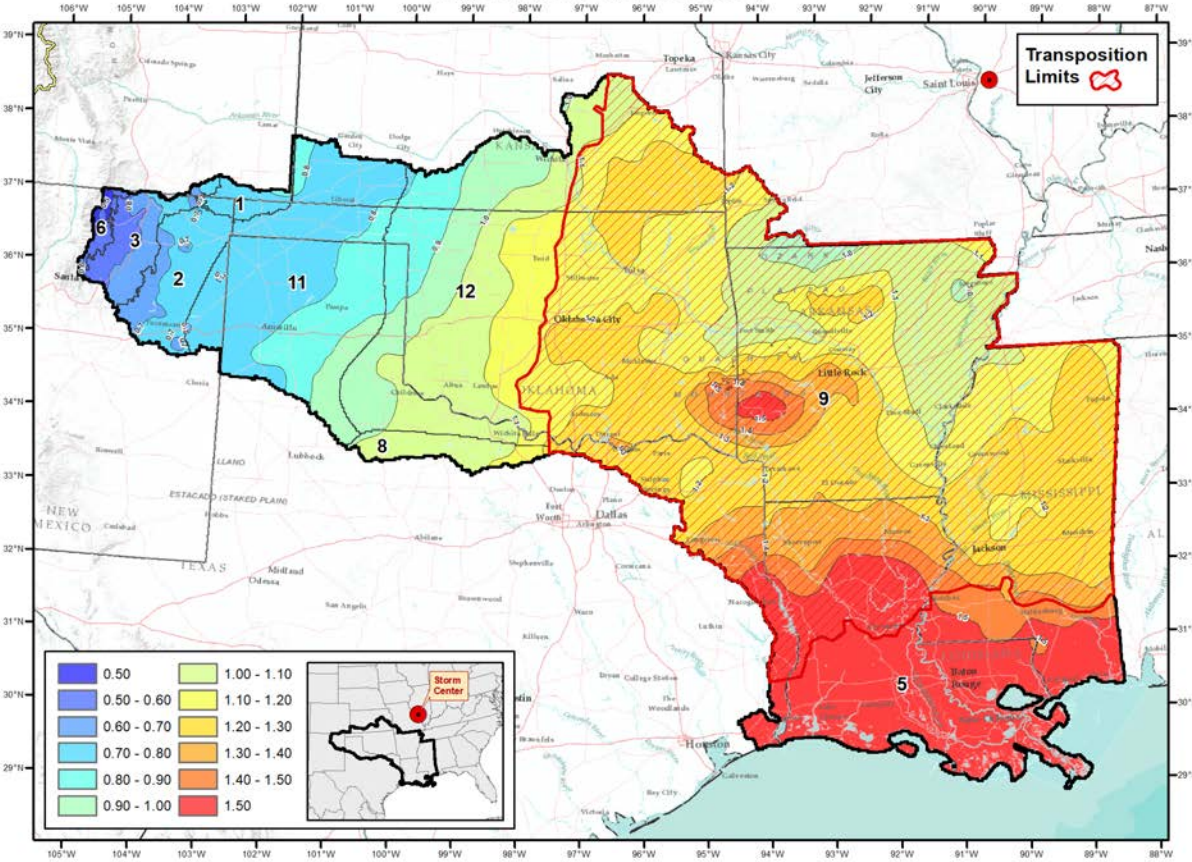
Gridded GTF for Elba, AL
Mar, 1929 (SPAS 1305 DAD zone 1)



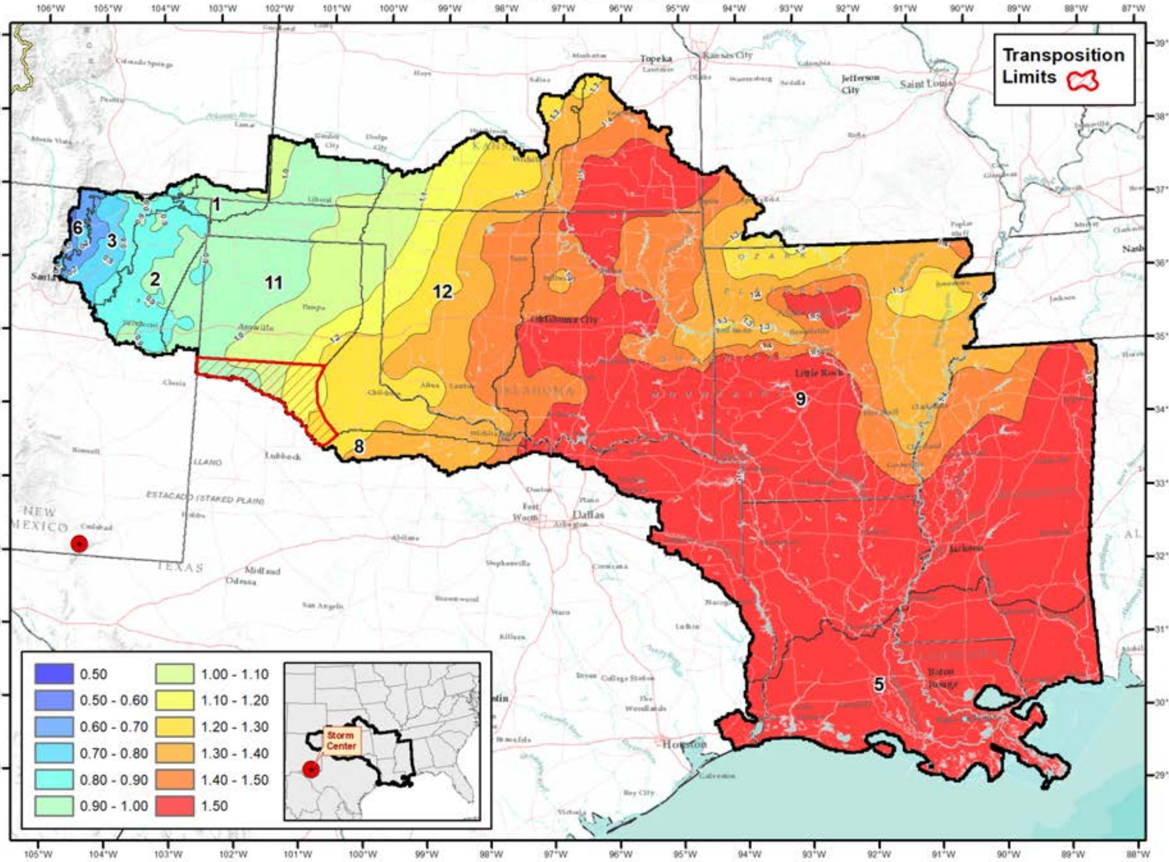
Gridded GTF for Warner, OK
May, 1943 (SPAS 1431 DAD zone 1)



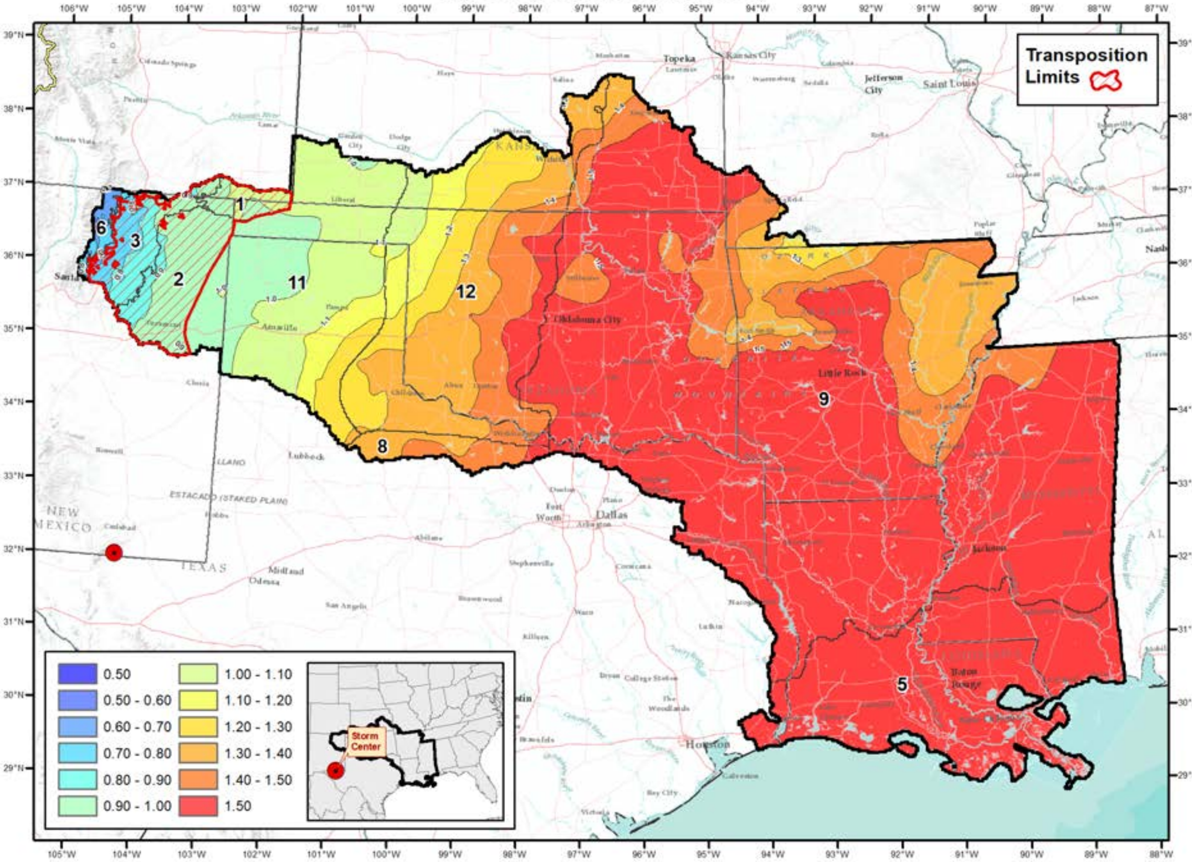
Gridded GTF for Collinsville, IL
Aug, 1946 (SPAS 1433 DAD zone 1)



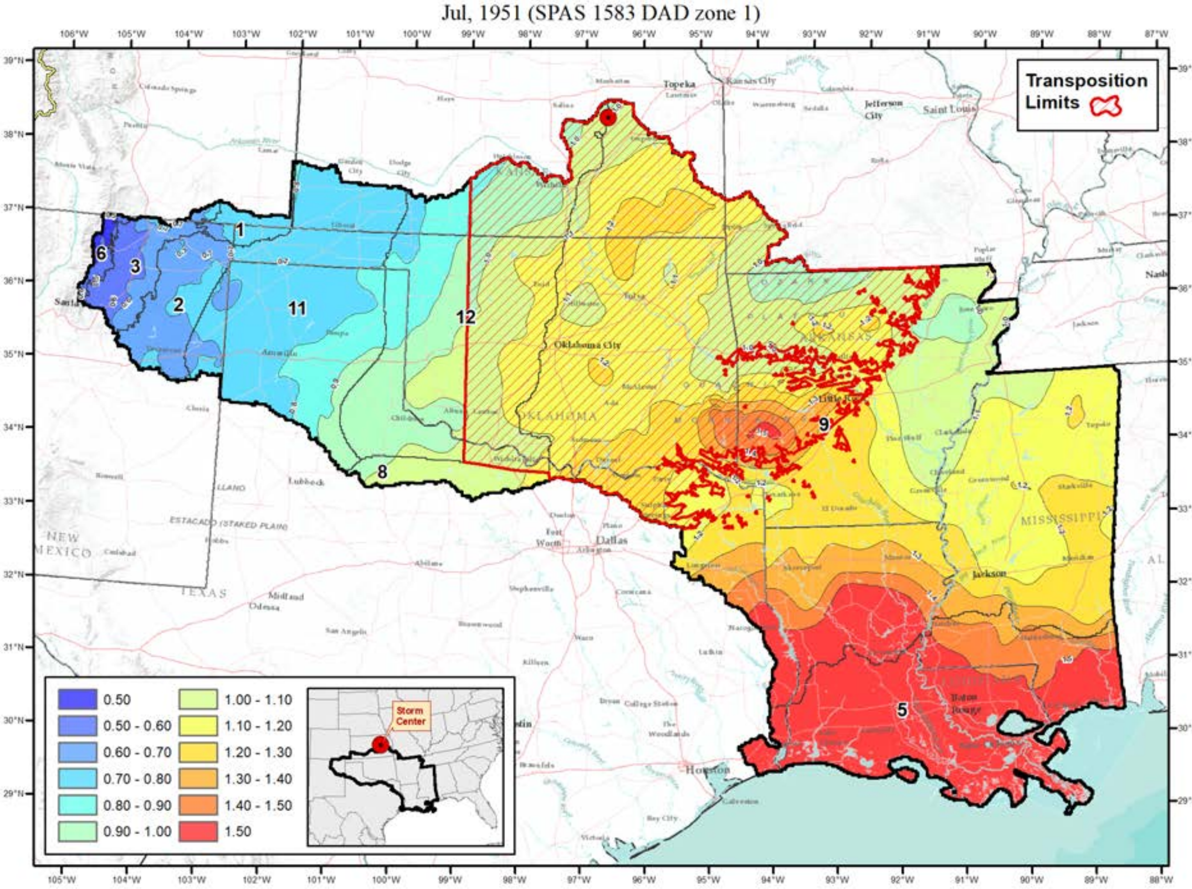
Gridded GTF for Mccolleum Ranch, NM
Sep, 1941 (SPAS 1486 DAD zone 1)



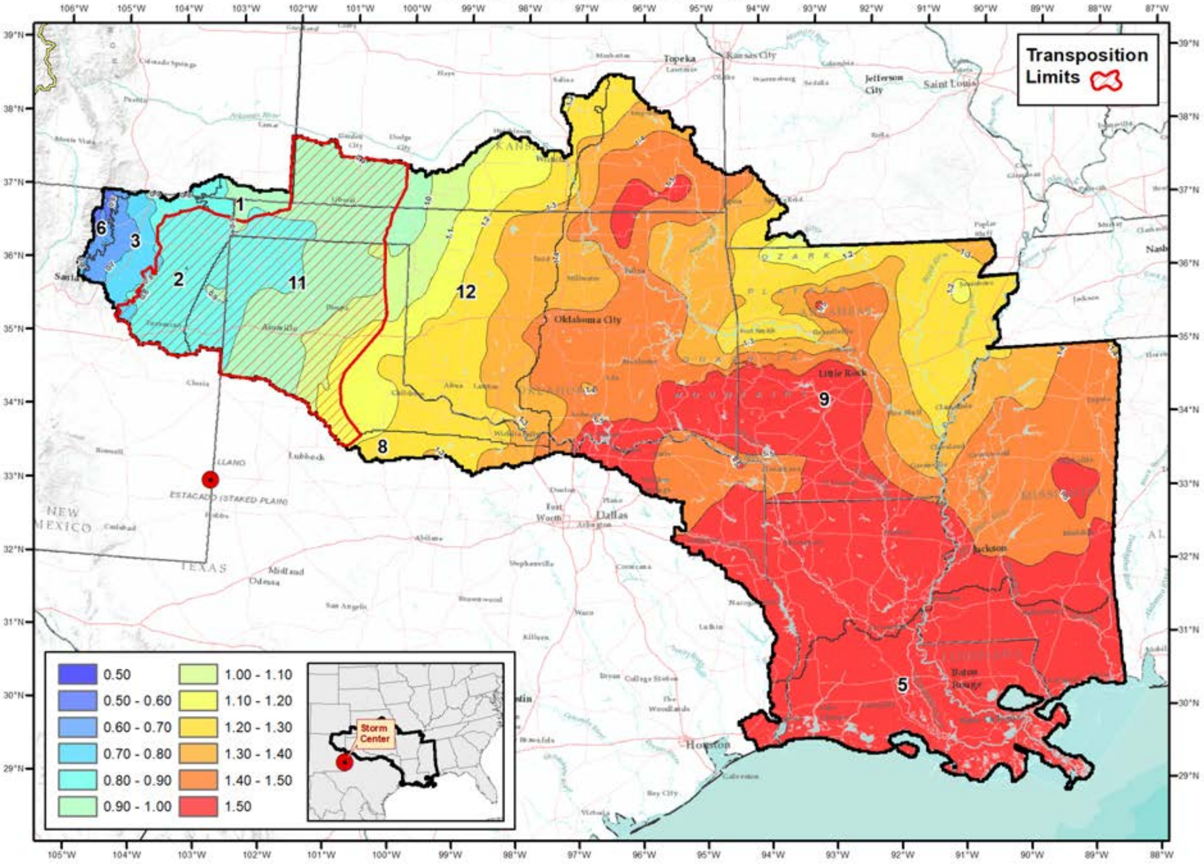
Gridded GTF for Guadalupe Pass, TX
Sep, 2013 (SPAS 1530 DAD zone 1)



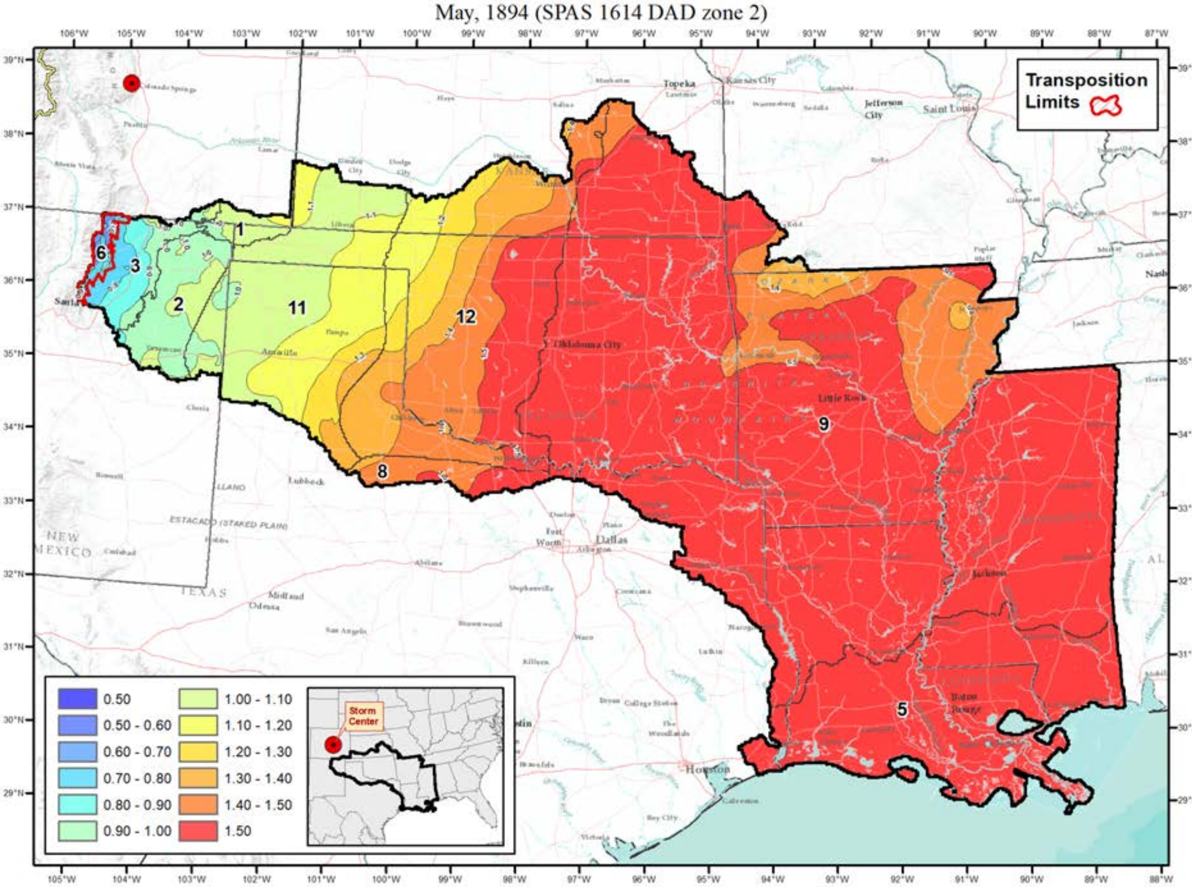
Gridded GTF for Council Grove, KS
Jul, 1951 (SPAS 1583 DAD zone 1)



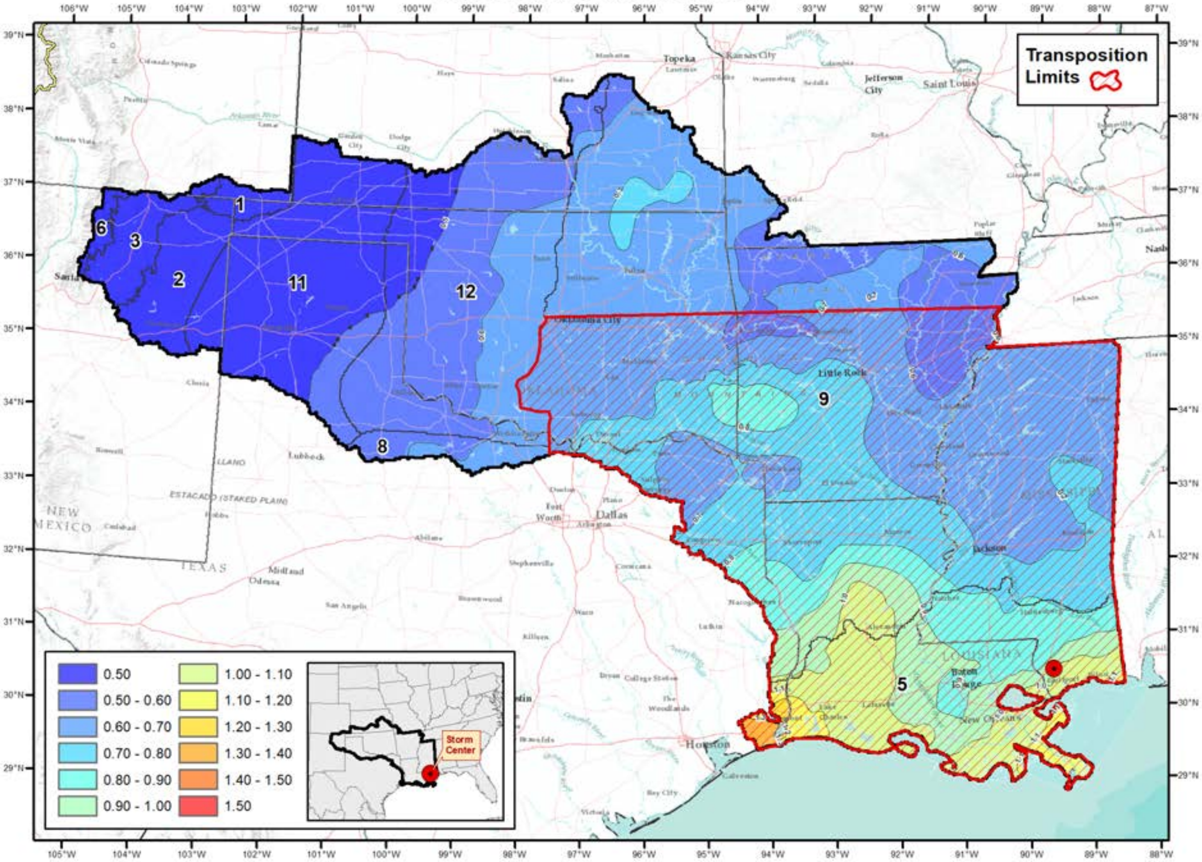
Gridded GTF for Prairieview, NM
May, 1941 (SPAS 1587 DAD zone 1)



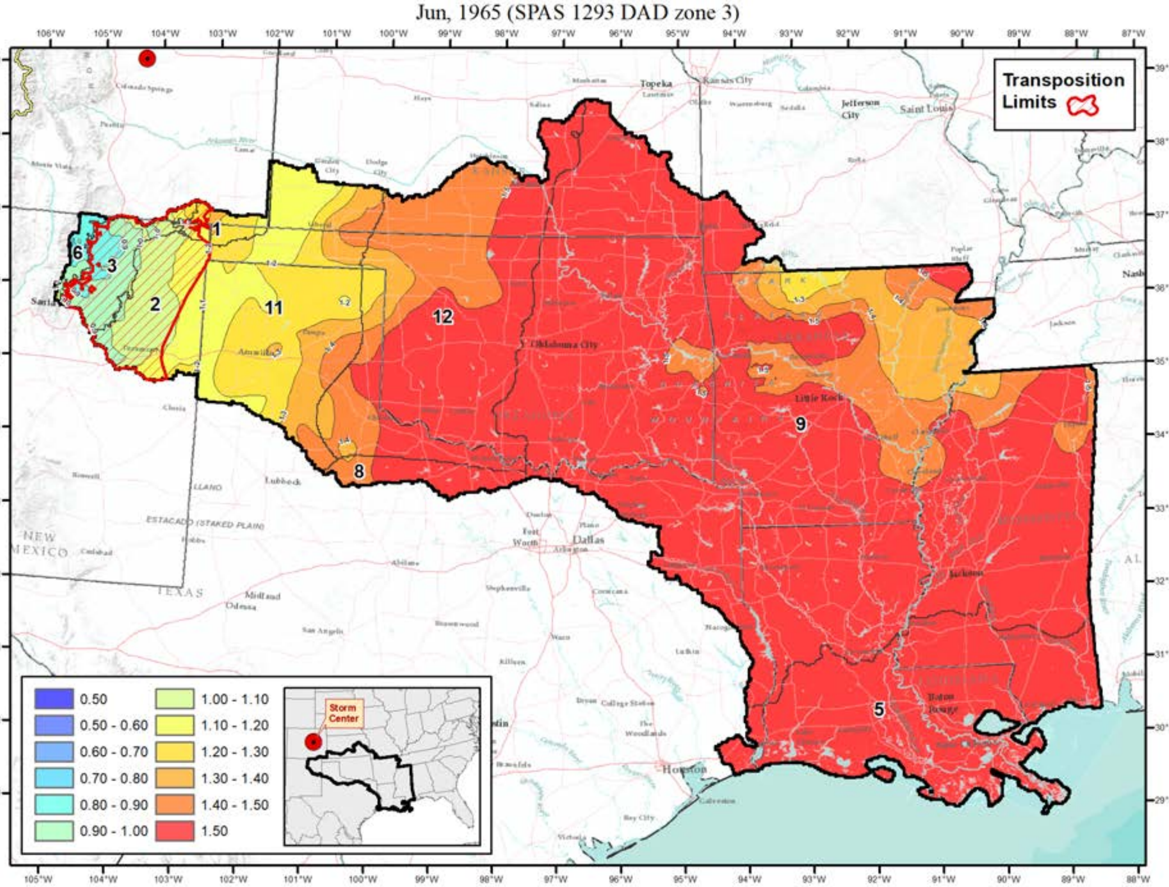
Gridded GTF for Lake Moraine, CO
May, 1894 (SPAS 1614 DAD zone 2)



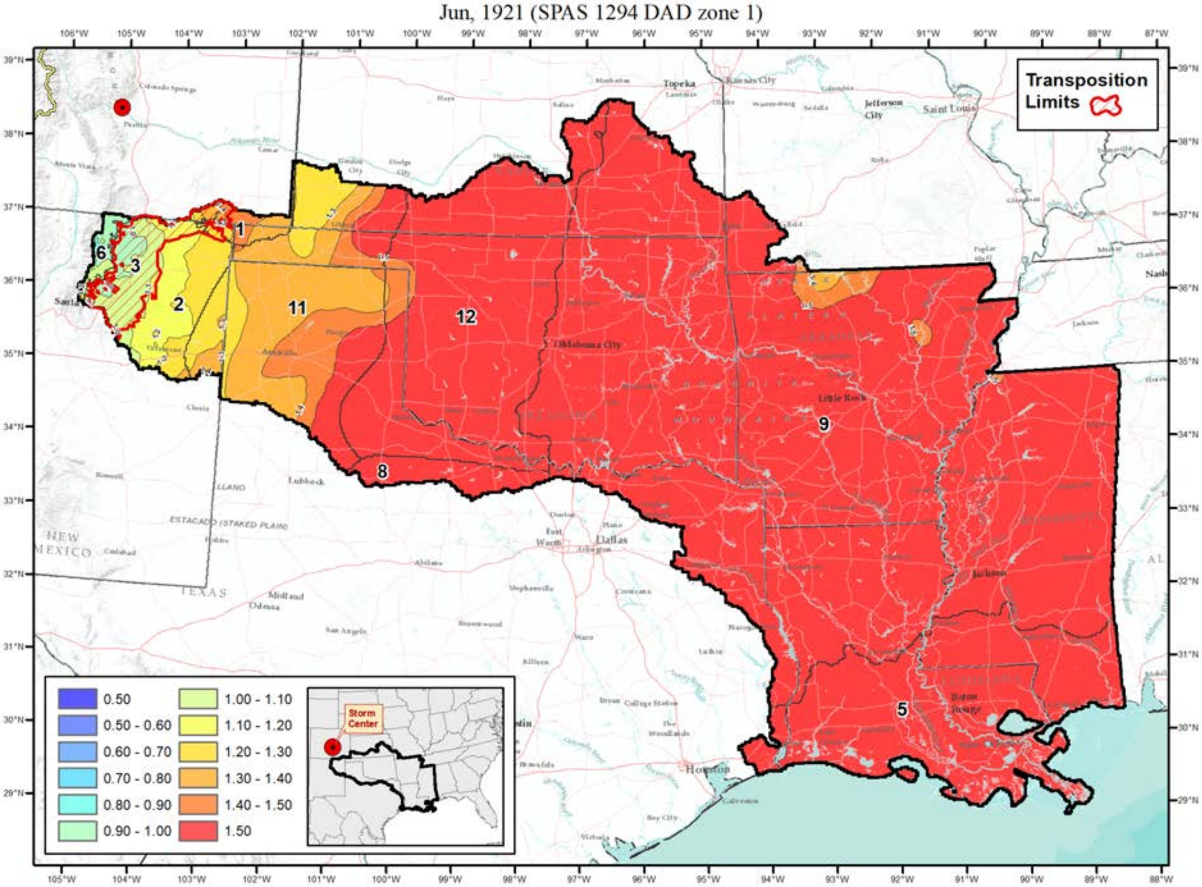
Gridded GTF for Necaise, LA
May, 1995 (SPAS 1719 DAD zone 1)



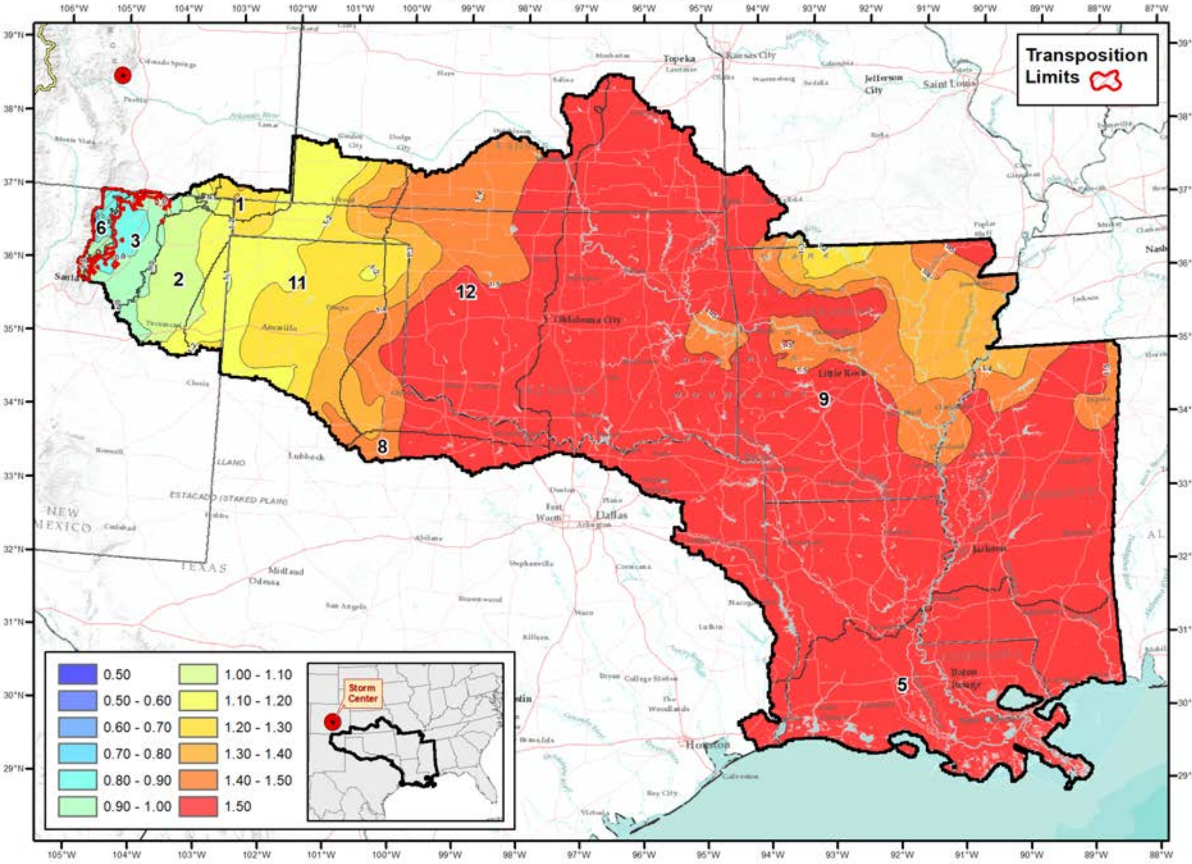
Gridded GTF for Elbert, CO
Jun, 1965 (SPAS 1293 DAD zone 3)



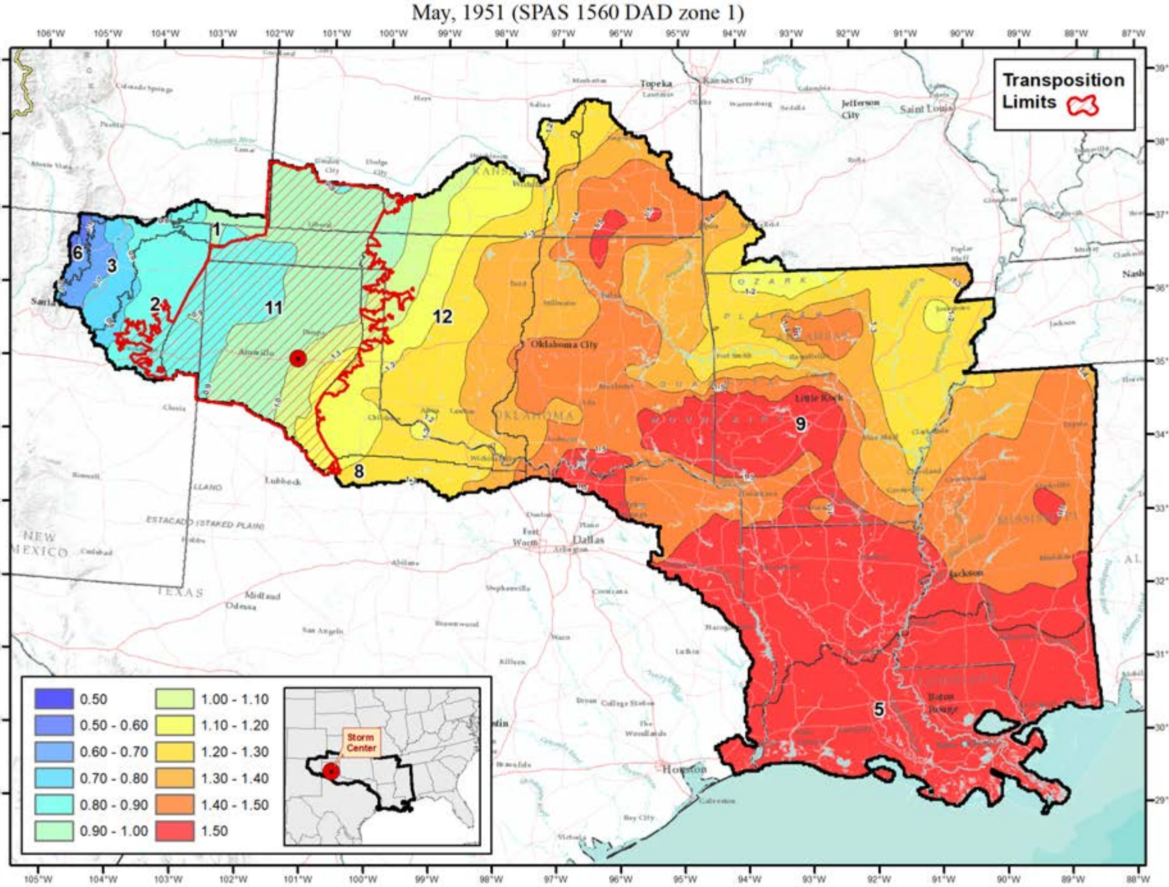
Gridded GTF for Penrose, CO
Jun, 1921 (SPAS 1294 DAD zone 1)



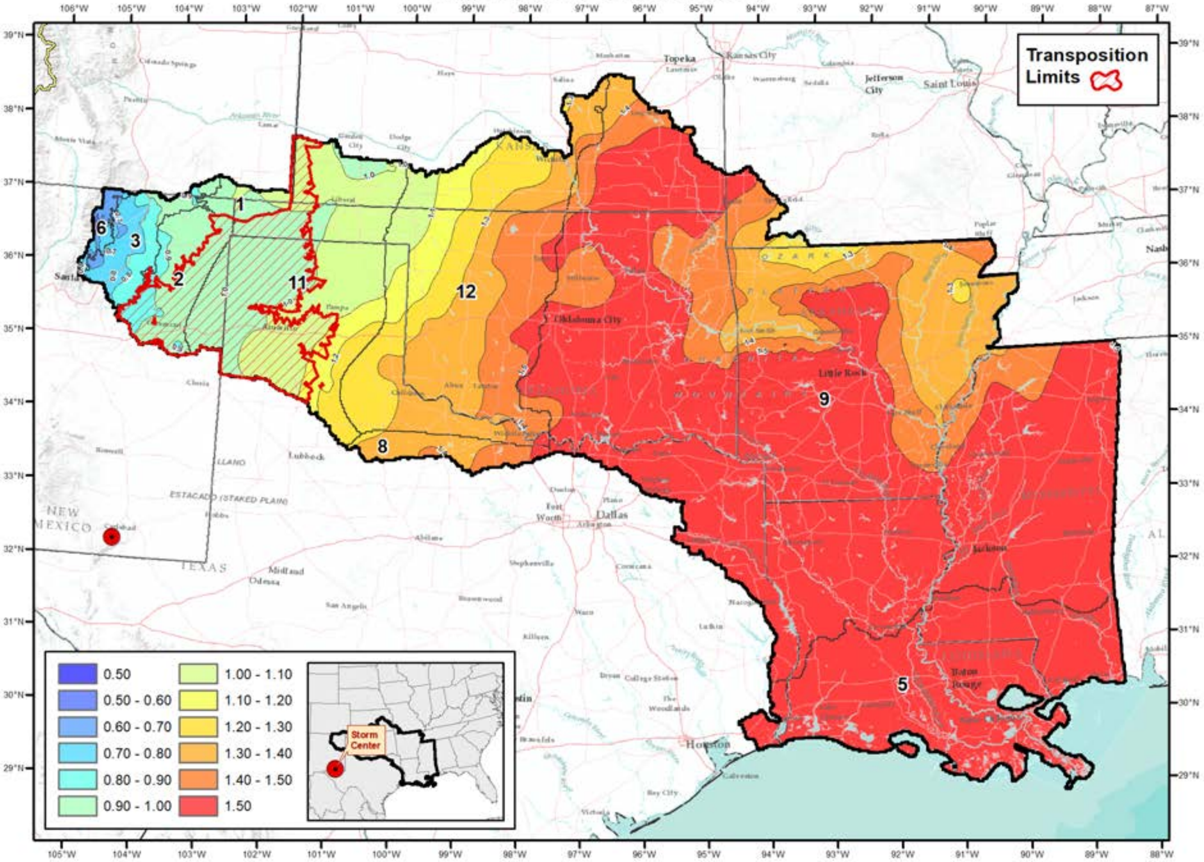
Gridded GTF for Adelaide, CO
Jun, 1921 (SPAS 1294 DAD zone 2)



Gridded GTF for Conway, TX
May, 1951 (SPAS 1560 DAD zone 1)

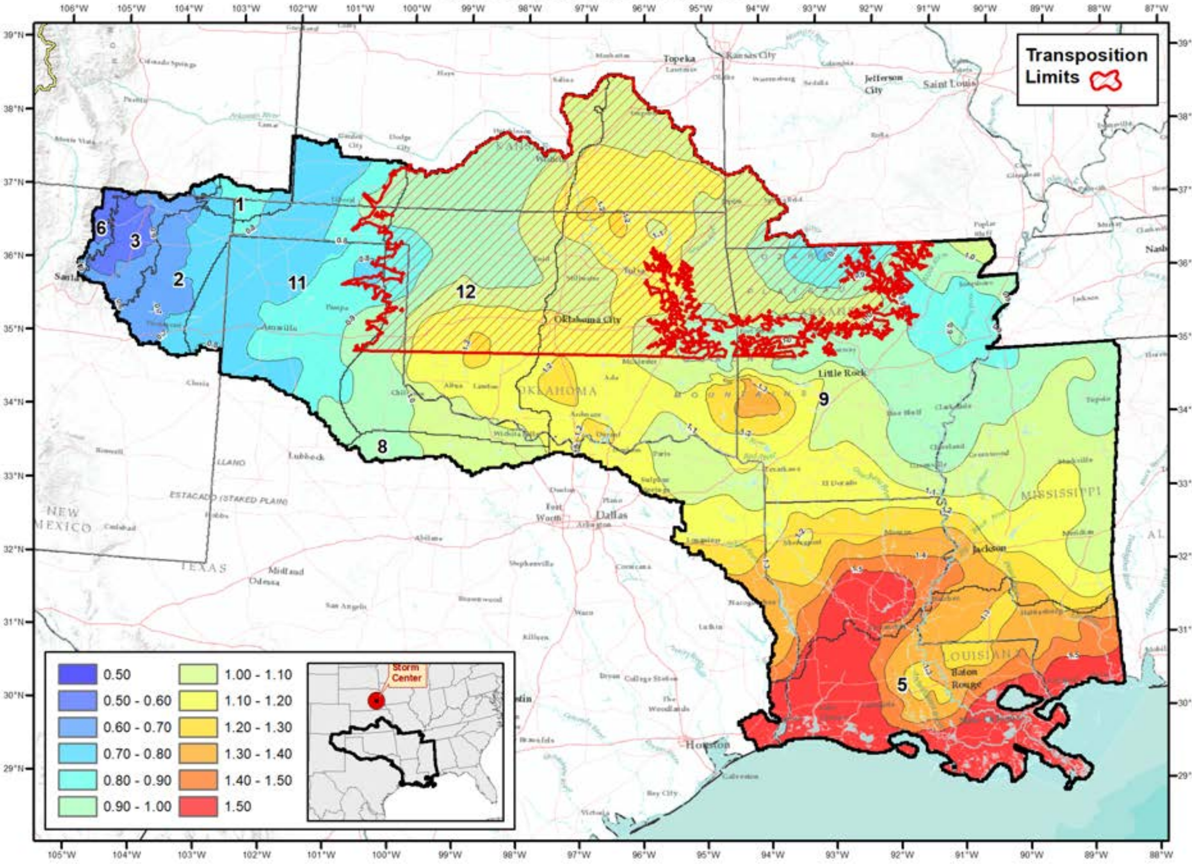


Gridded GTF for Carlsbad, NM
Aug, 1966 (SPAS 1568 DAD zone 1)

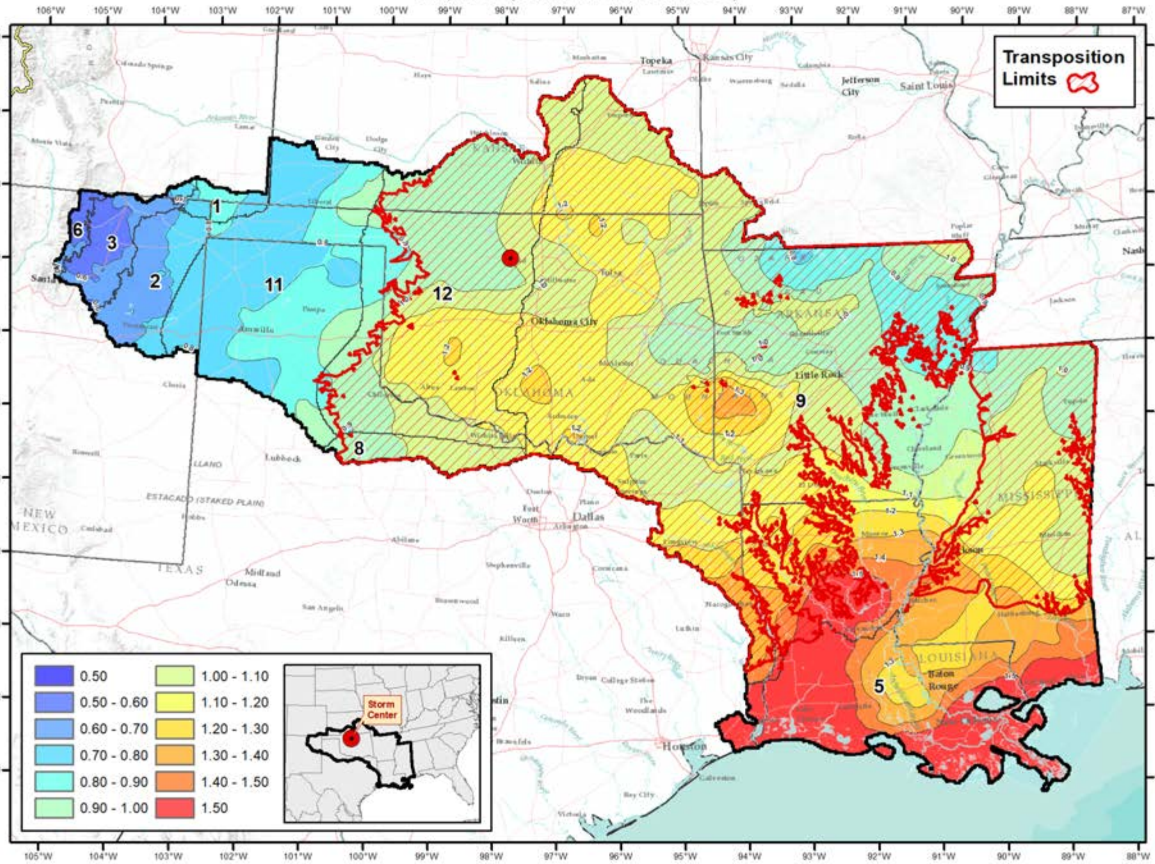


# Local Storms

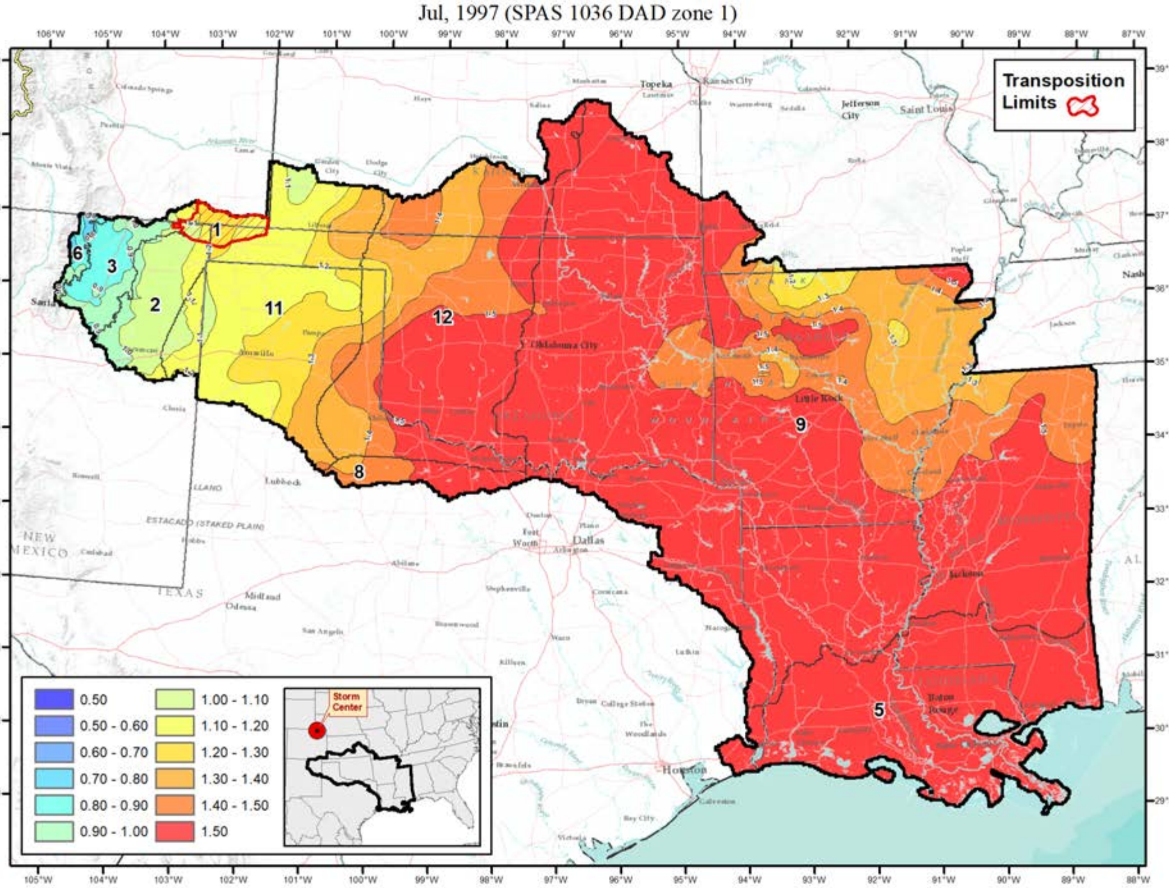
Gridded GTF for David City, NE
Jun, 1963 (SPAS 1030 DAD zone 1)



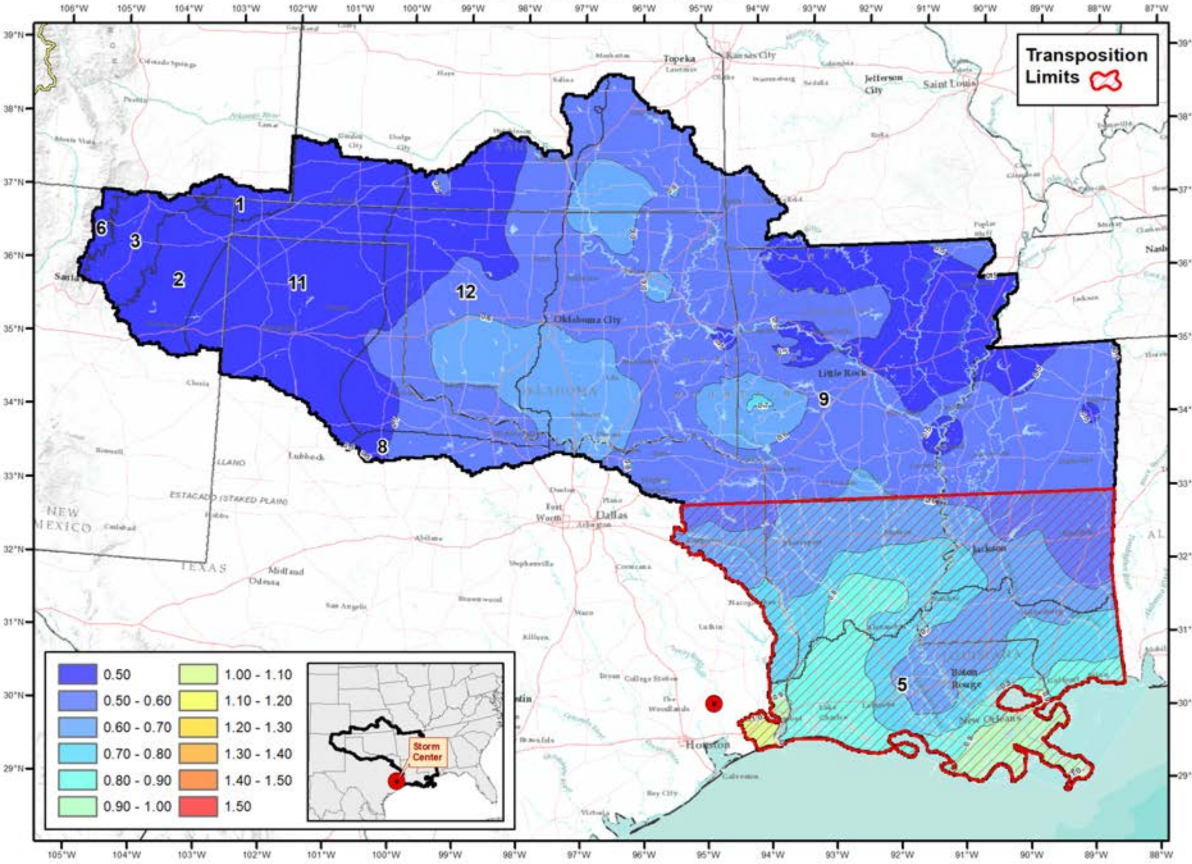
Gridded GTF for Enid, OK
Oct, 1973 (SPAS 1034 DAD zone 1)



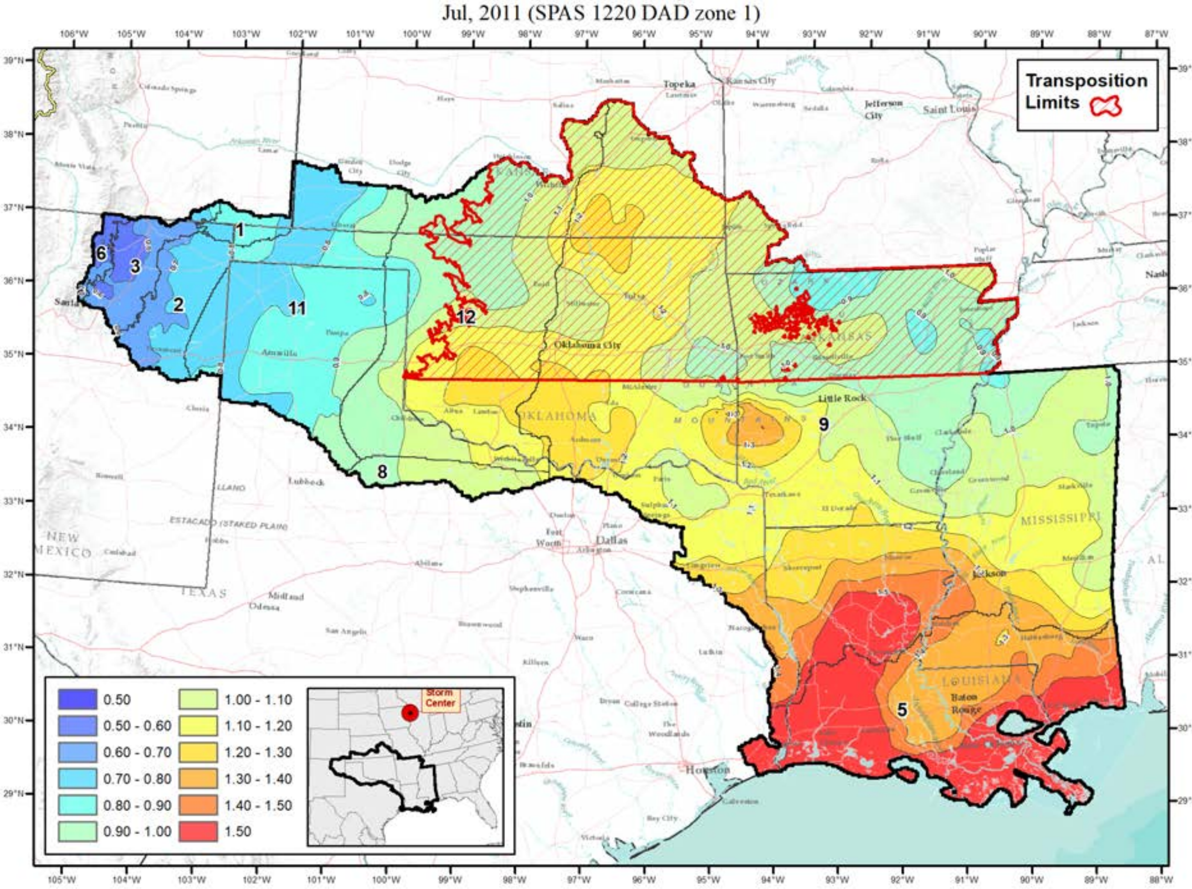
Gridded GTF for Pawnee Creek, CO
Jul, 1997 (SPAS 1036 DAD zone 1)



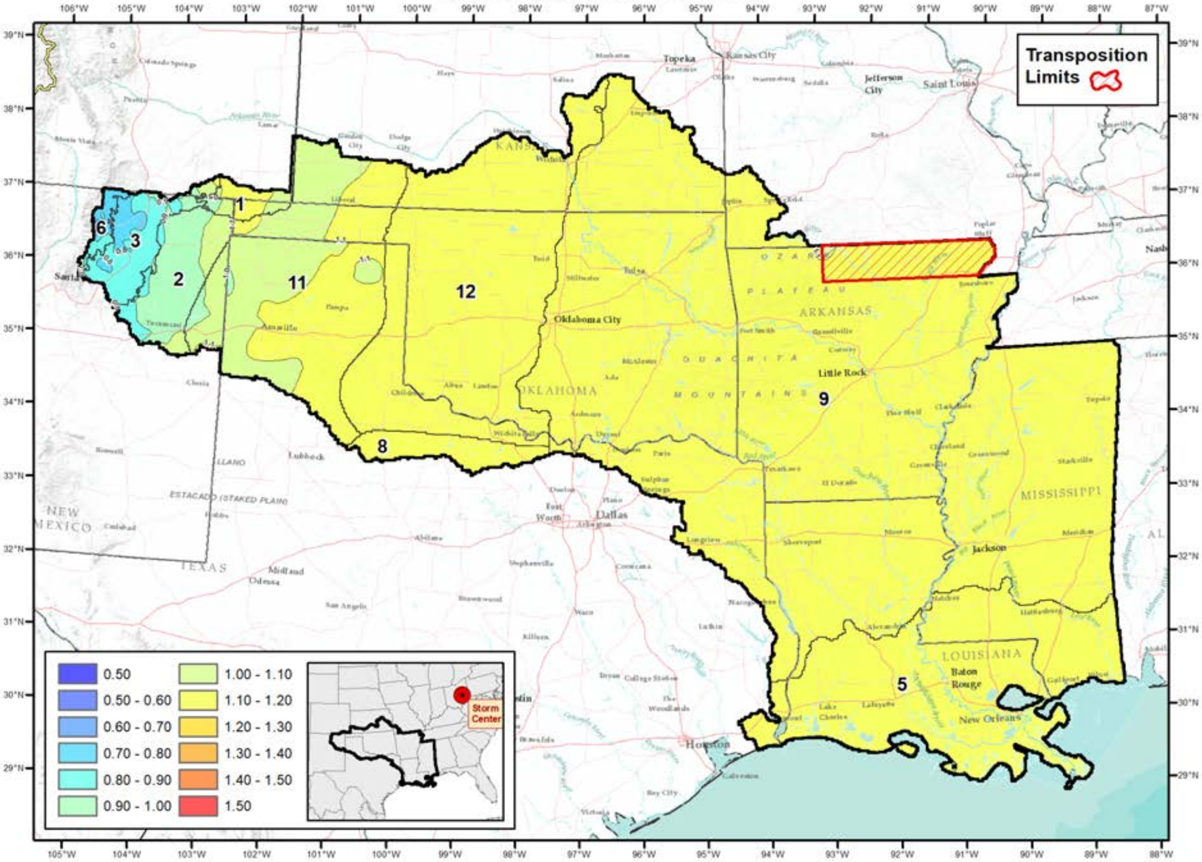
Gridded GTF for Corrigan, TX
Oct, 1994 (SPAS 1185 DAD zone 1)



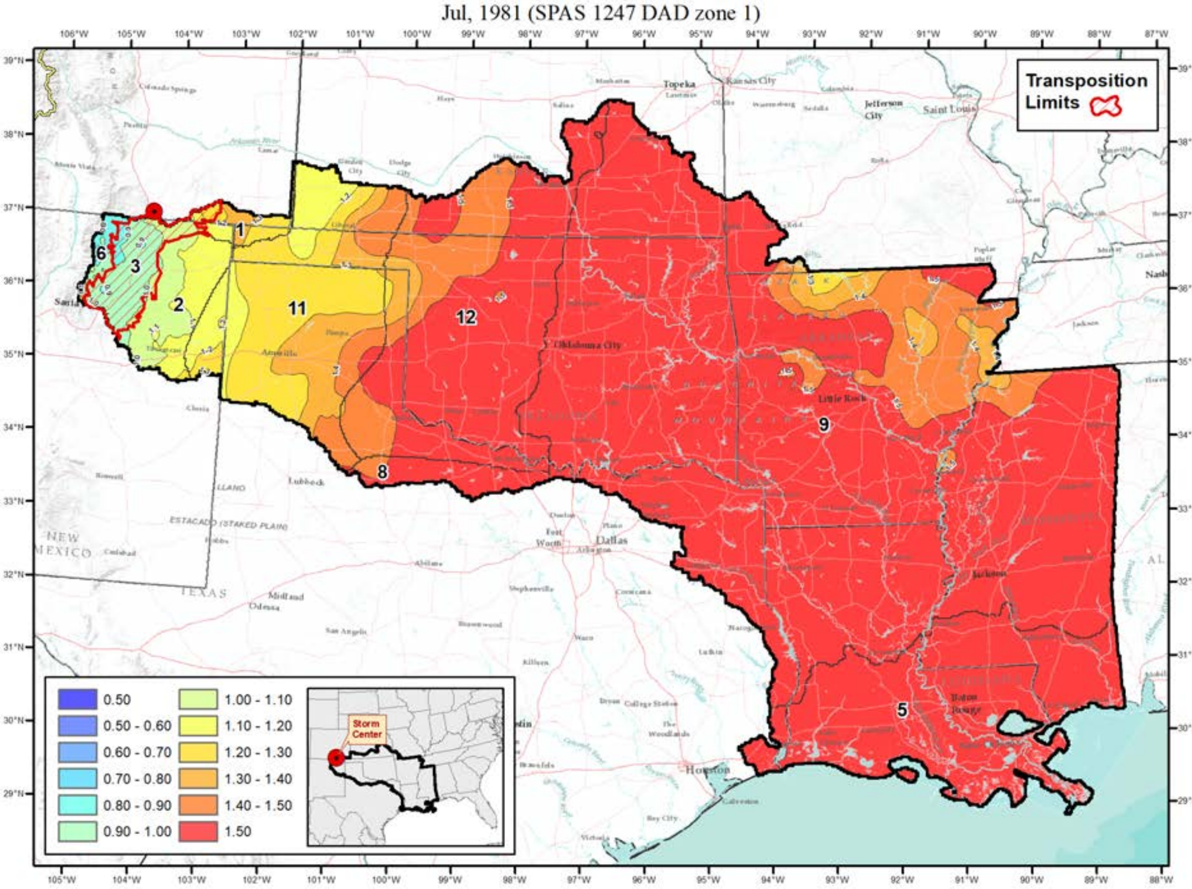
Gridded GTF for Dubuque, IA
Jul, 2011 (SPAS 1220 DAD zone 1)



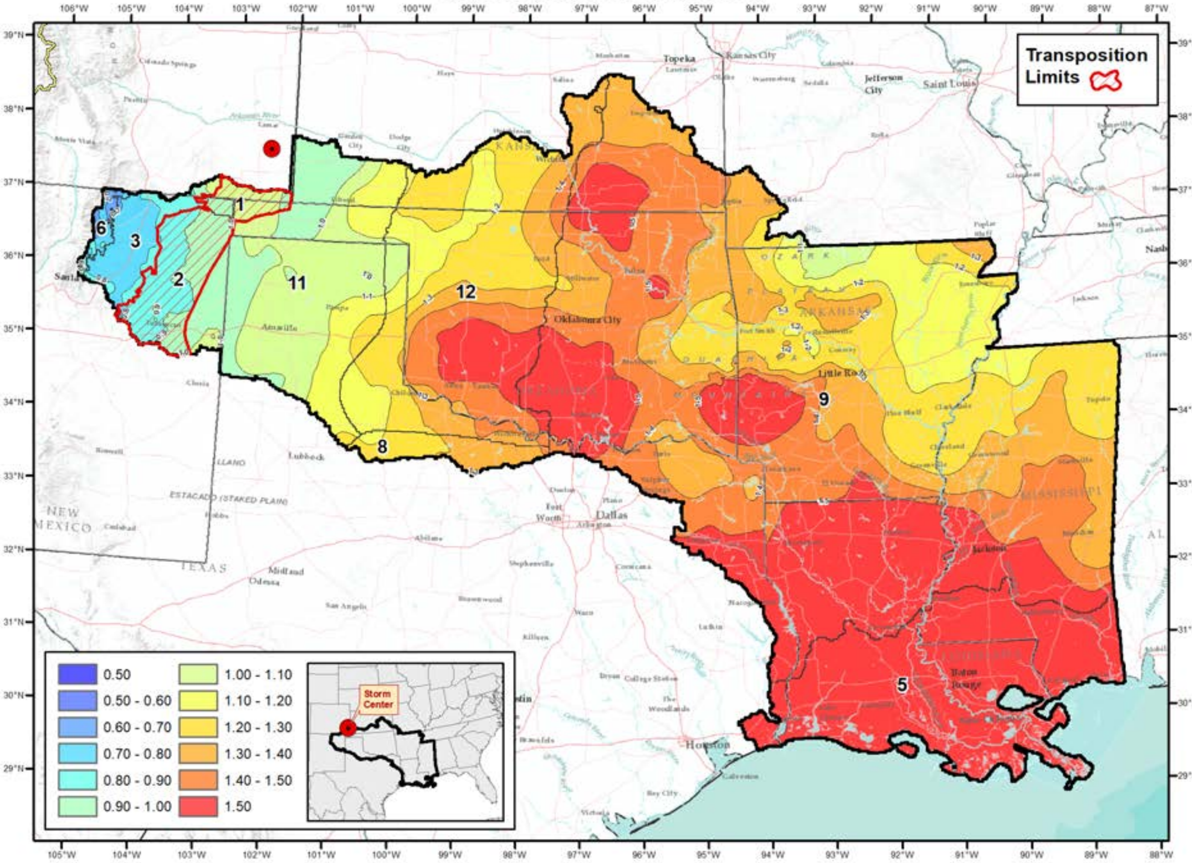
Gridded GTF for College Hill, OH
Jun, 1963 (SPAS 1226 DAD zone 1)



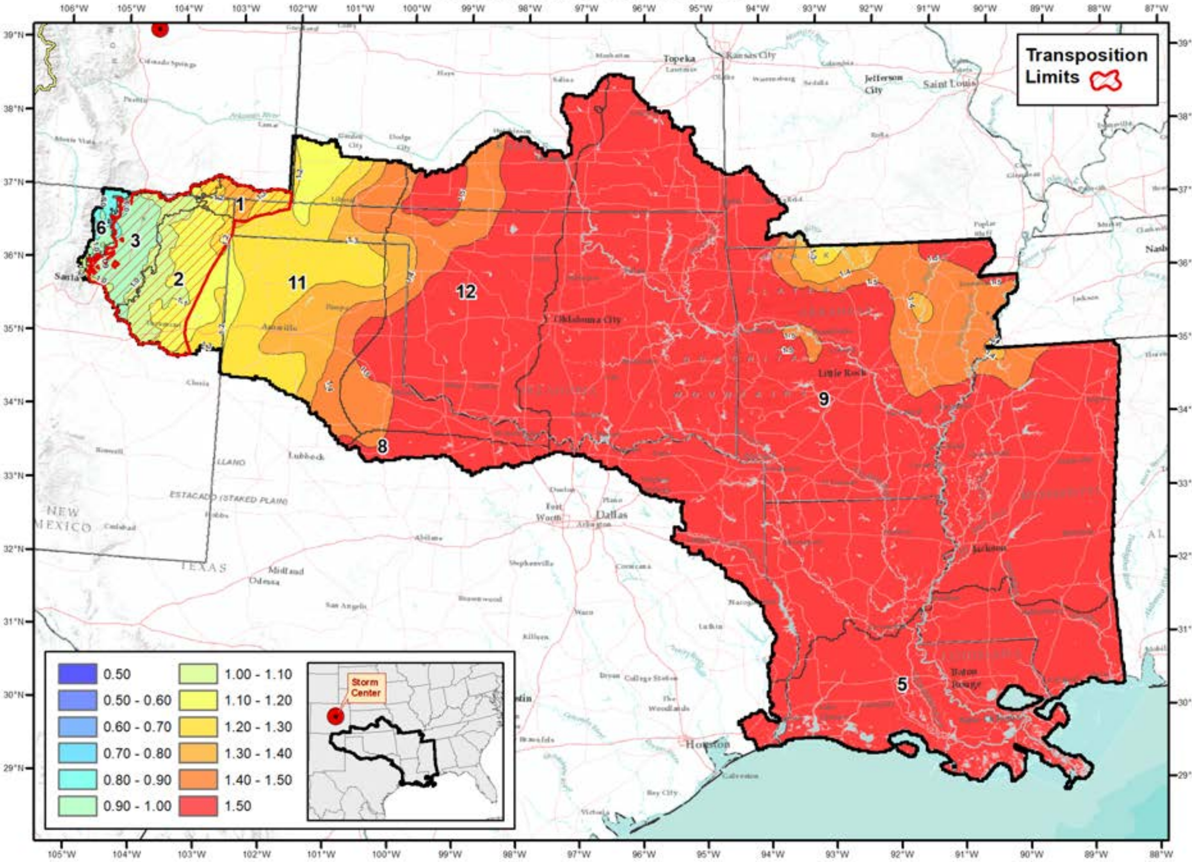
Gridded GTF for Frijole Creek, CO
Jul, 1981 (SPAS 1247 DAD zone 1)



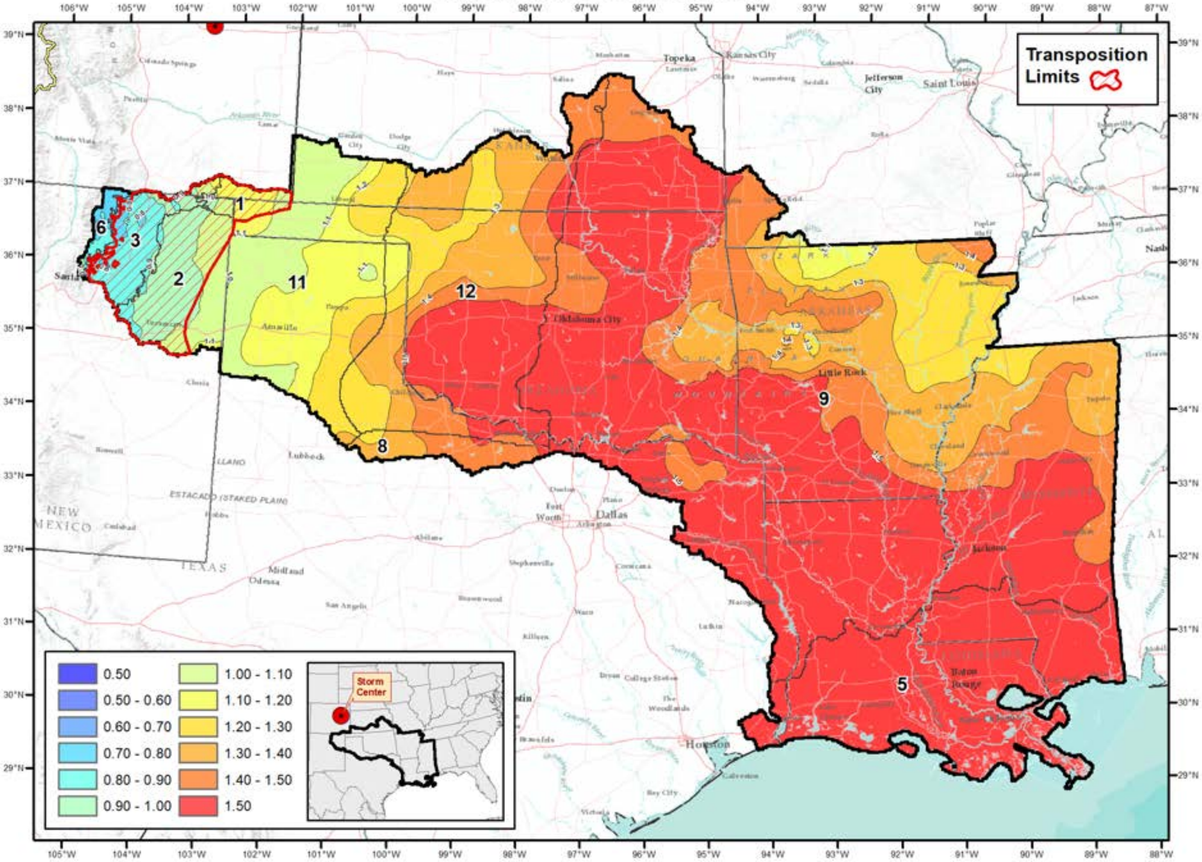
Gridded GTF for Holly, CO
Jun, 1965 (SPAS 1293 DAD zone 1)



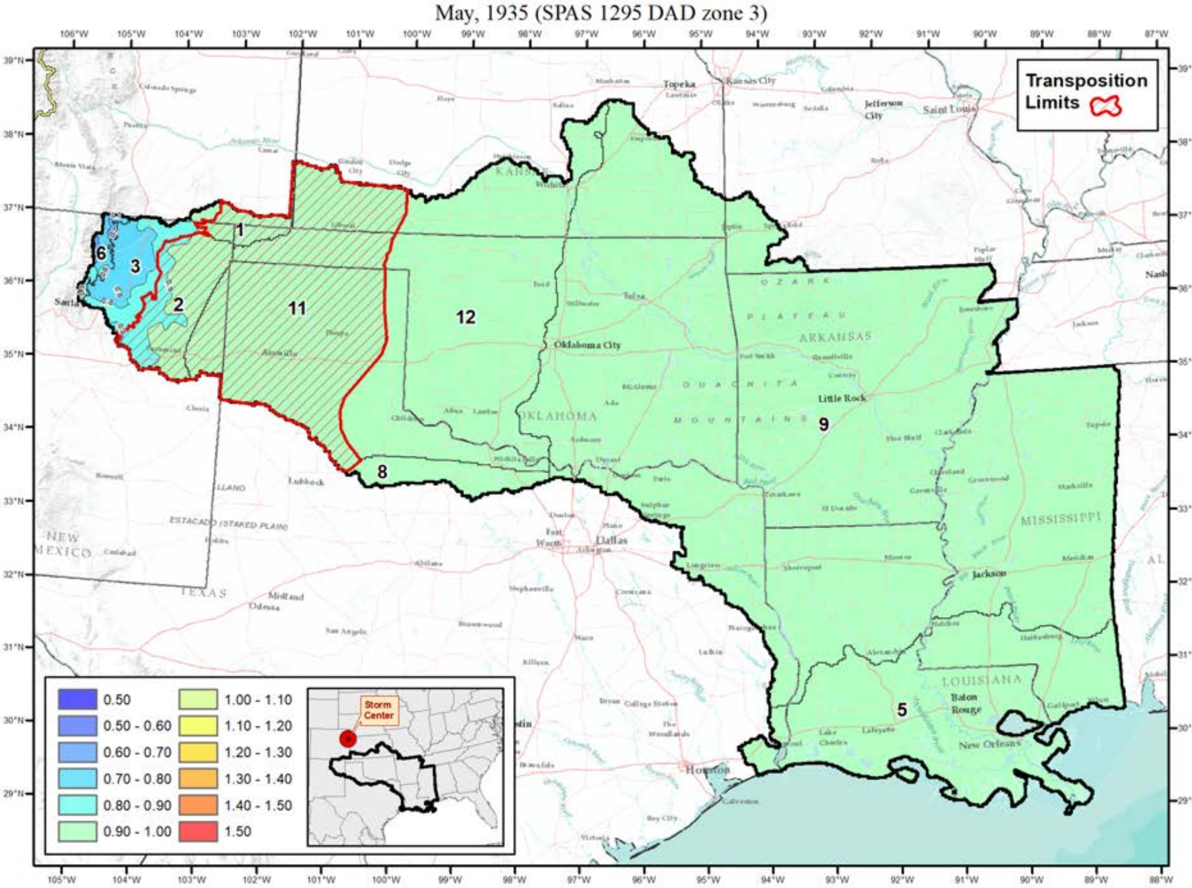
Gridded GTF for Elbert Cherry Creek, CO
May, 1935 (SPAS 1295 DAD zone 1)



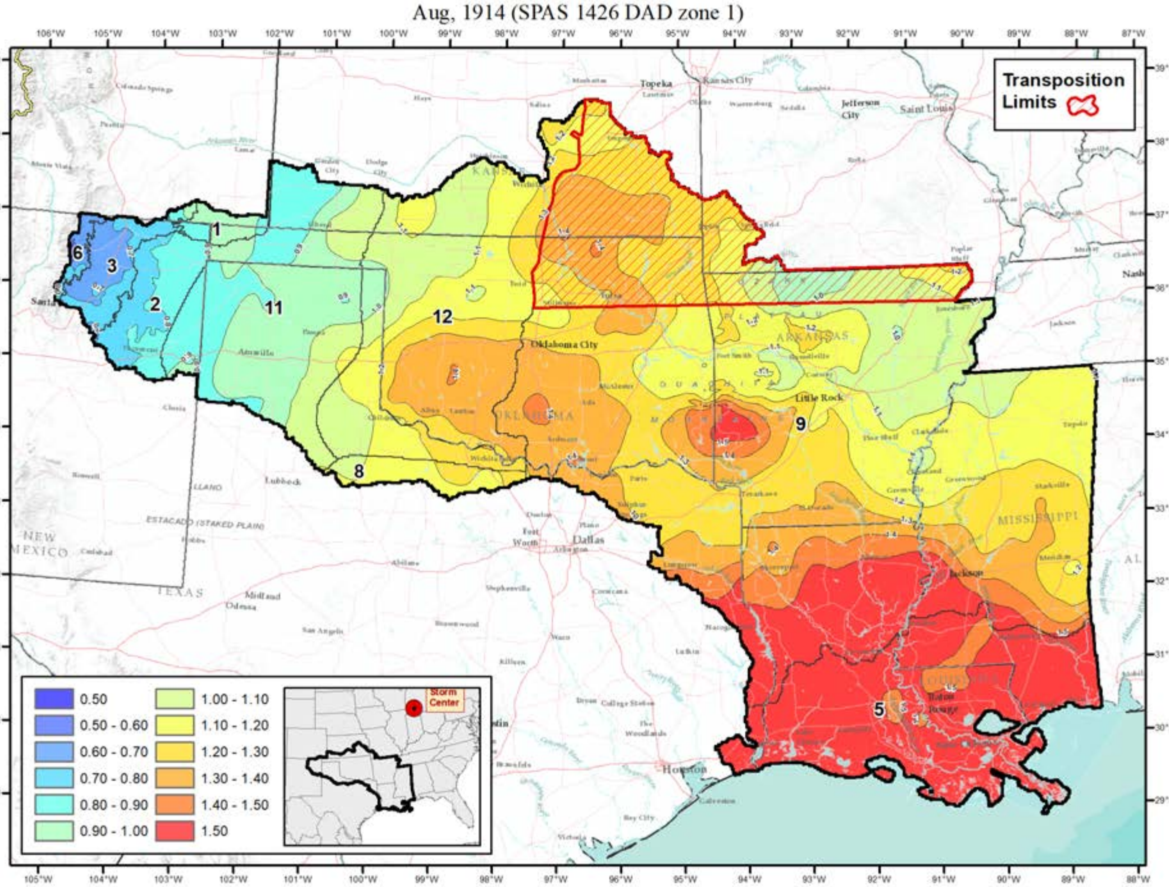
Gridded GTF for Genoa, CO
May, 1935 (SPAS 1295 DAD zone 2)



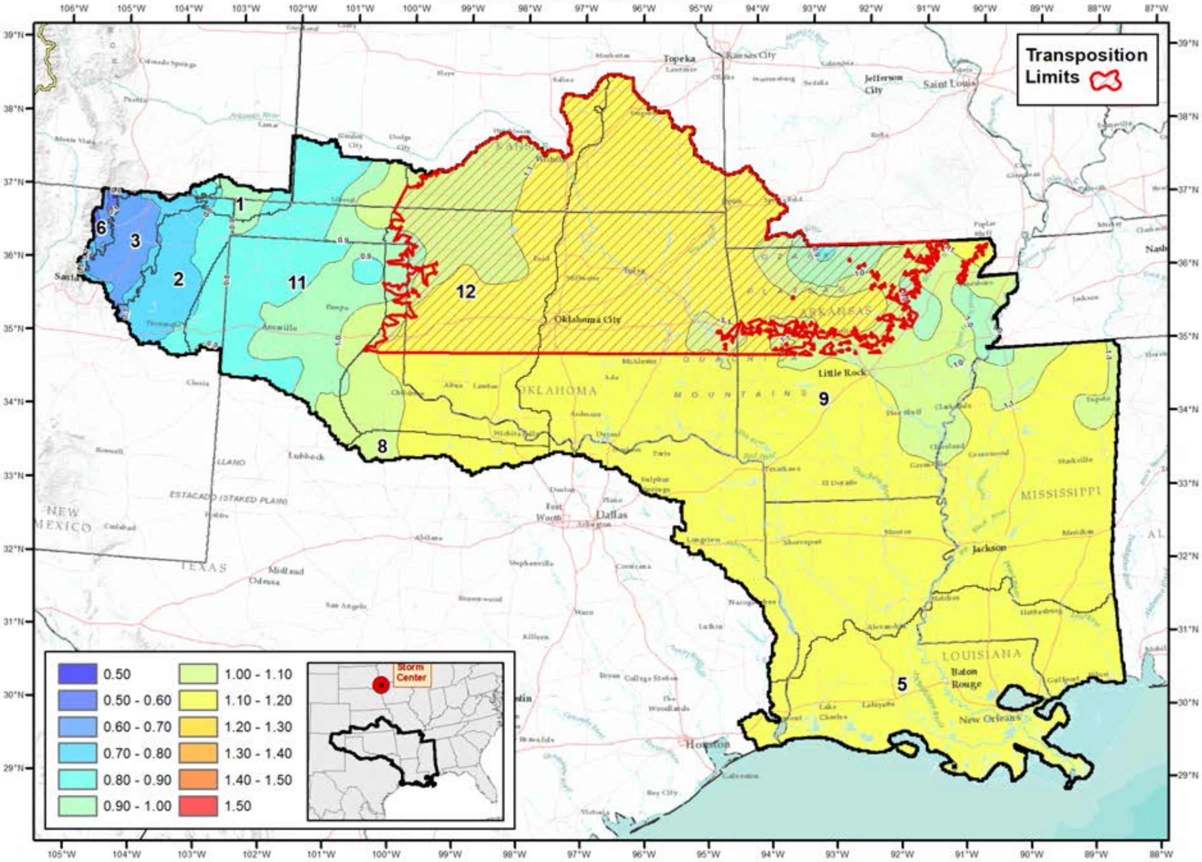
Gridded GTF for Hale, CO
May, 1935 (SPAS 1295 DAD zone 3)



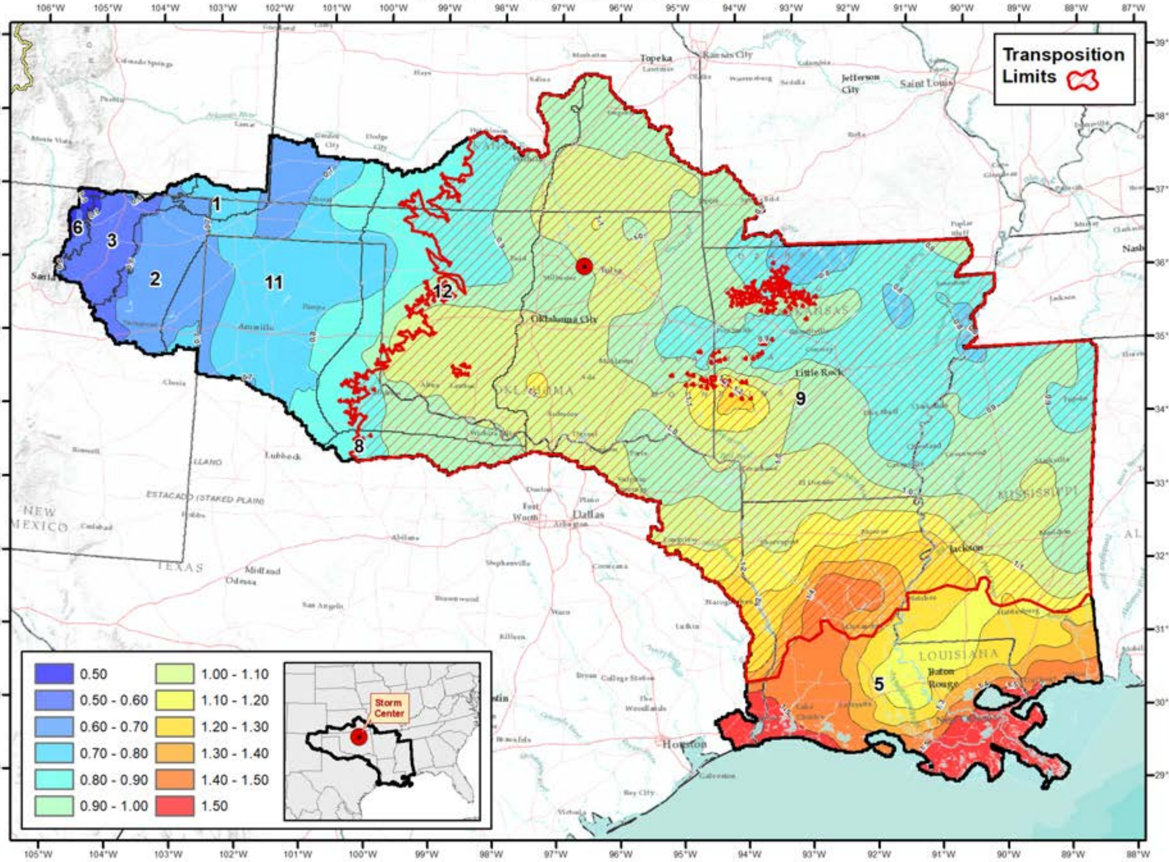
Gridded GTF for Cooper, MI
Aug, 1914 (SPAS 1426 DAD zone 1)



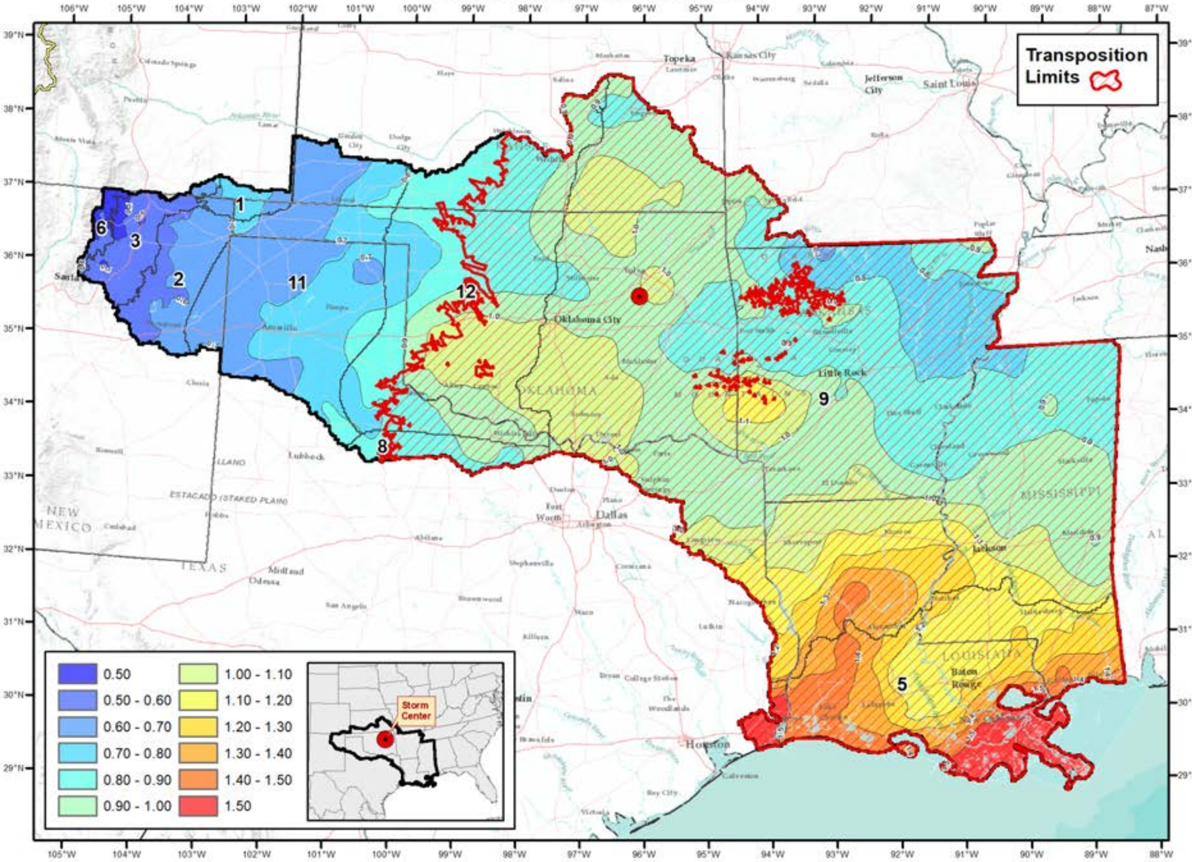
Gridded GTF for Boyden, IA
Sep, 1926 (SPAS 1427 DAD zone 1)



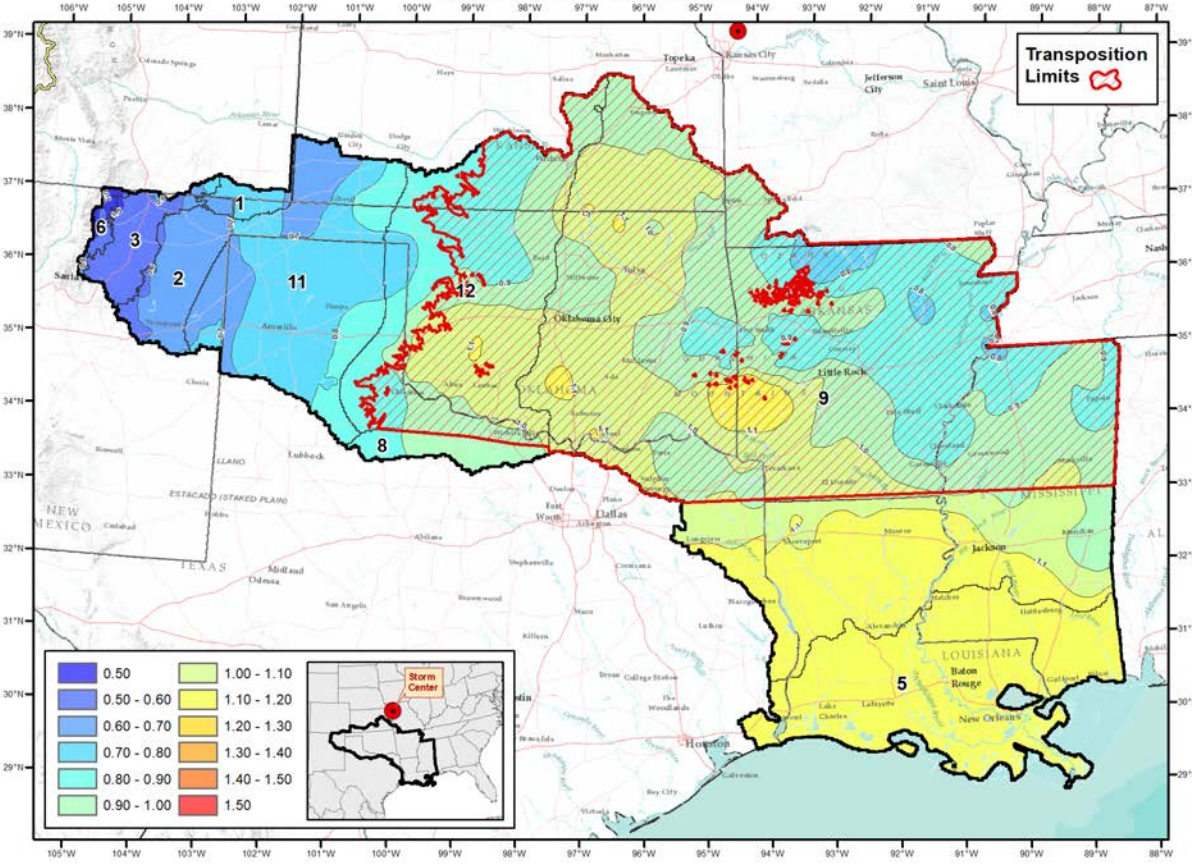
Gridded GTF for Hallett, OK
Sep, 1940 (SPAS 1429 DAD zone 2)



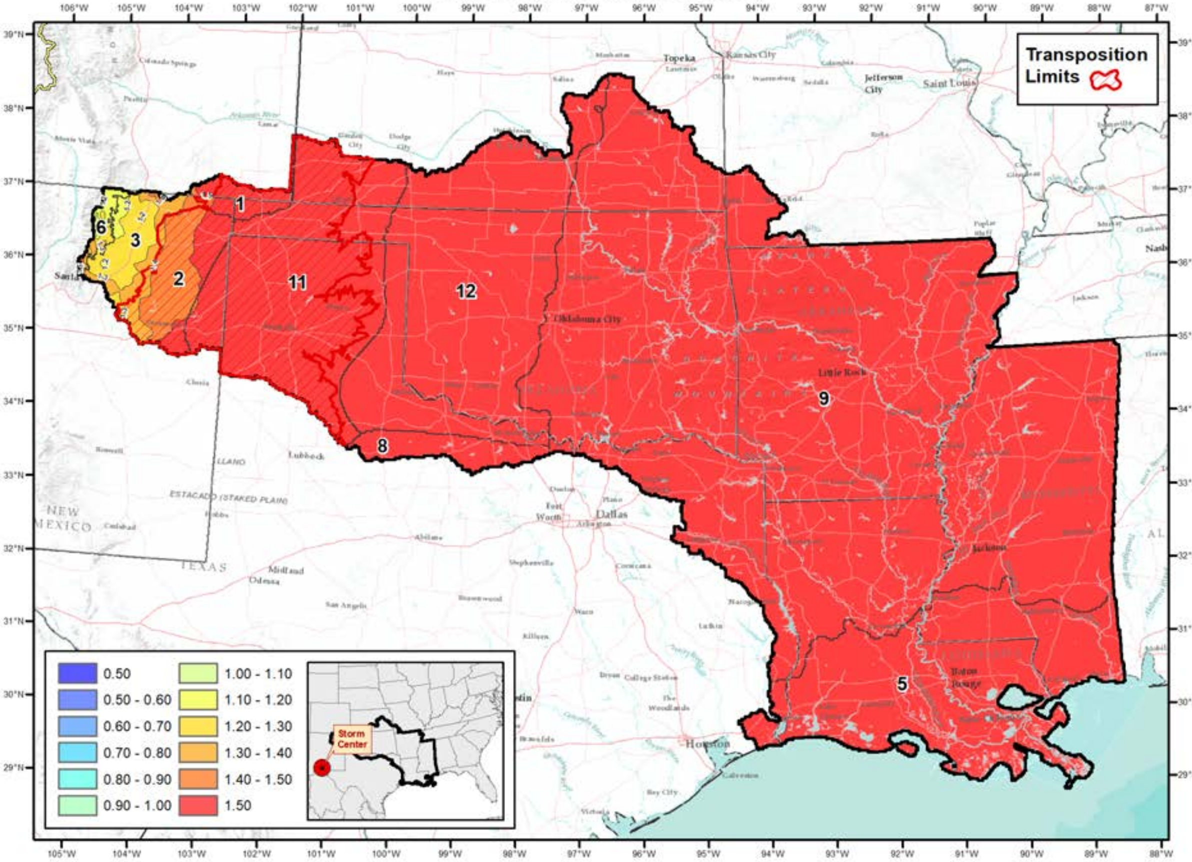
Gridded GTF for Mounds, OK
May, 1943 (SPAS 1432 DAD zone 1)



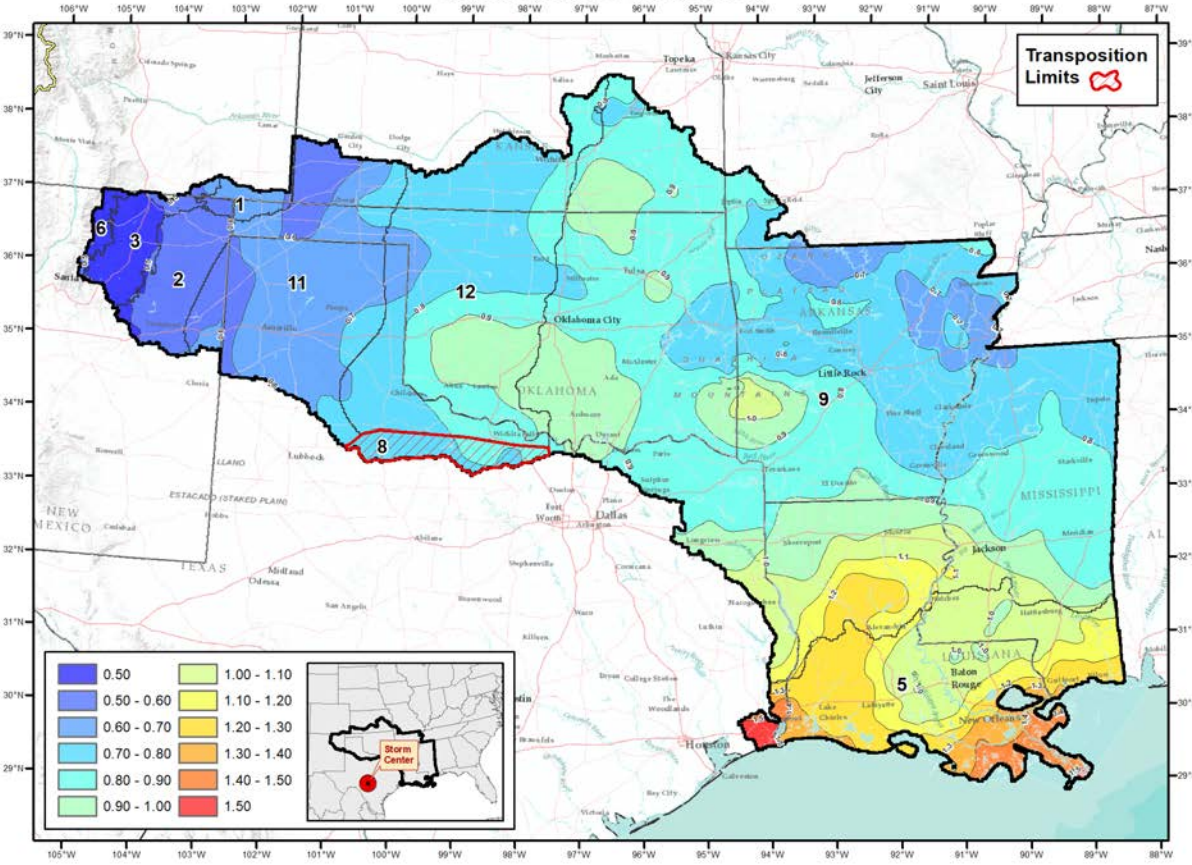
Gridded GTF for Holt, MO
Jun, 1947 (SPAS 1434 DAD zone 1)



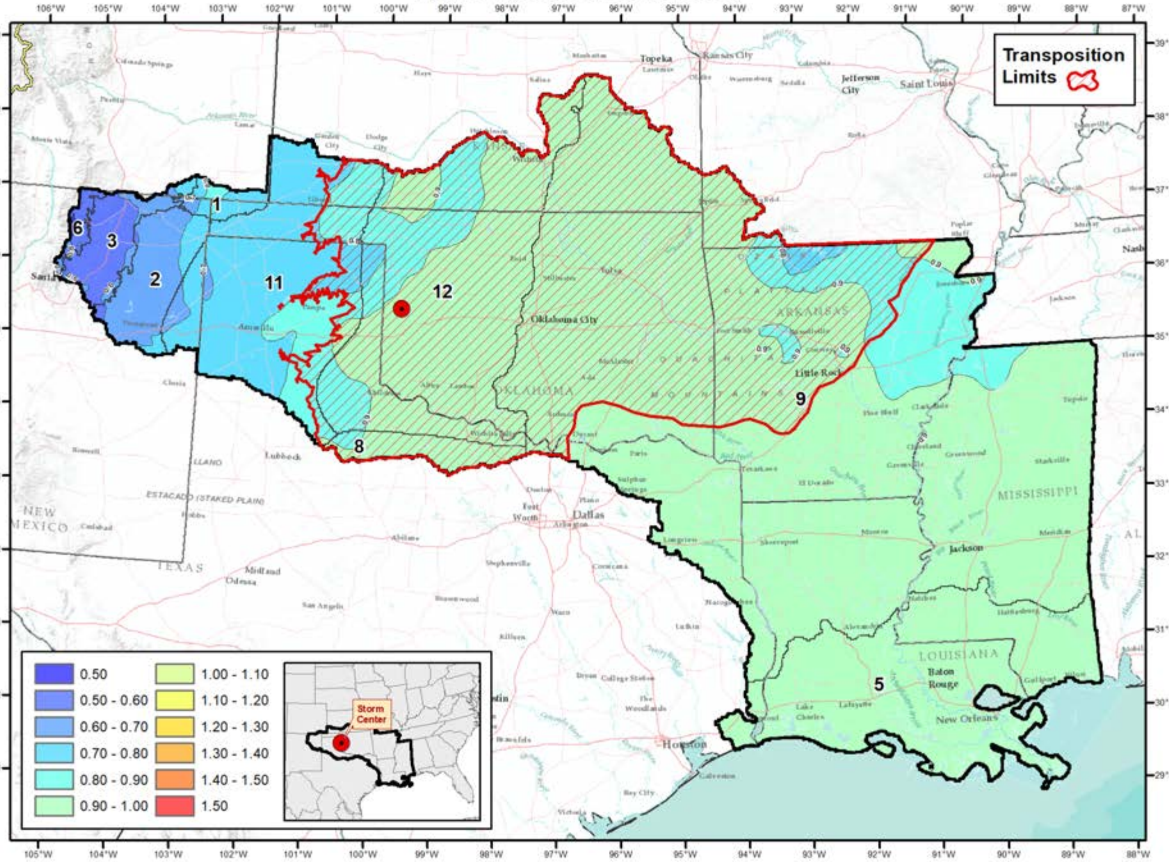
Gridded GTF for Las Cruces, NM
Aug, 1935 (SPAS 1485 DAD zone 1)



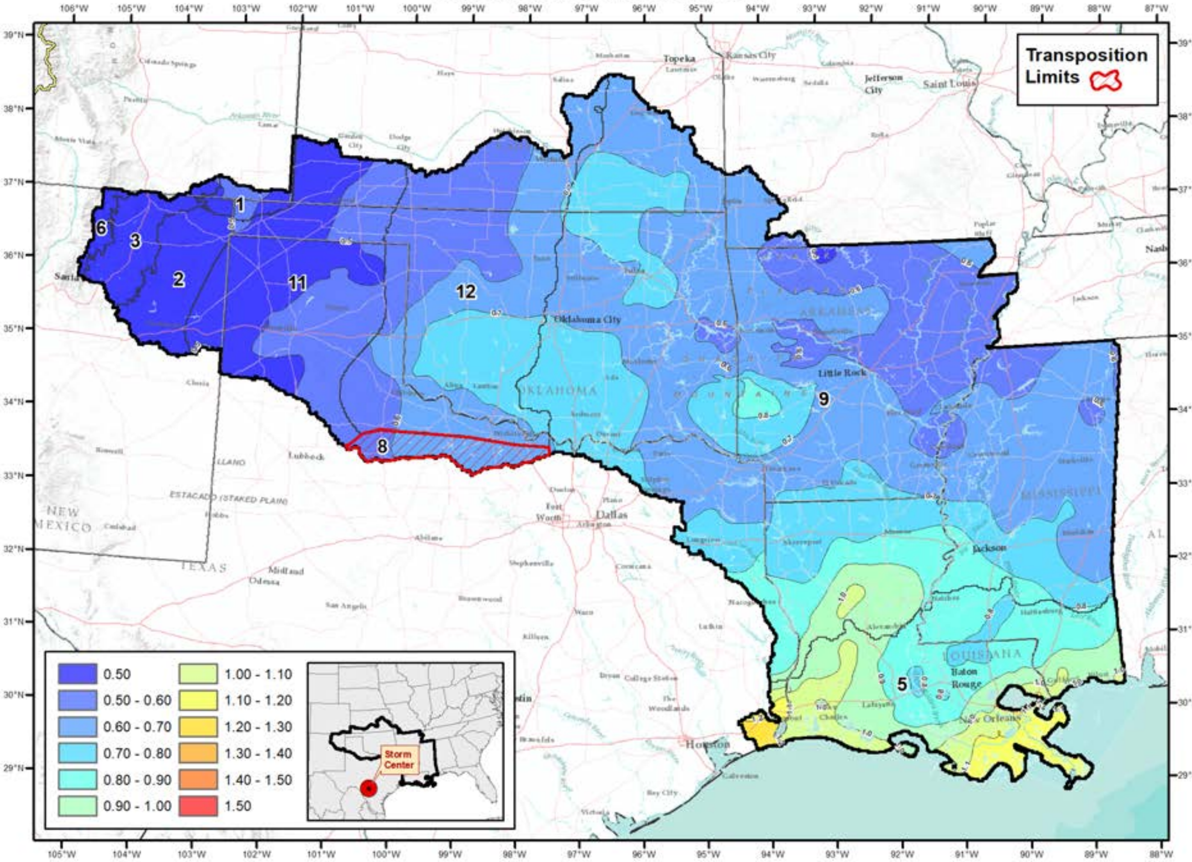
Gridded GTF for Mountain Home, TX
Jun, 1932 (SPAS 1494 DAD zone 1)



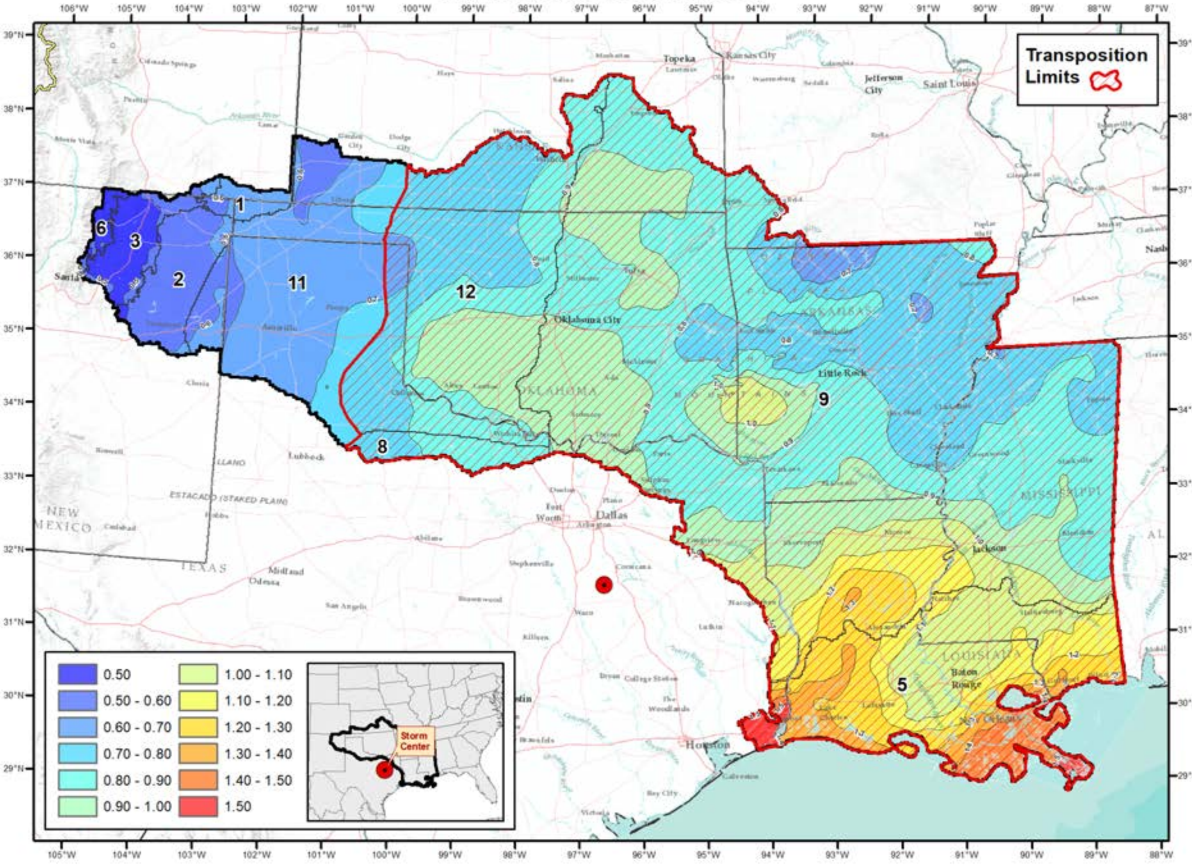
Gridded GTF for Cheyenne, OK
Apr, 1934 (SPAS 1495 DAD zone 1)



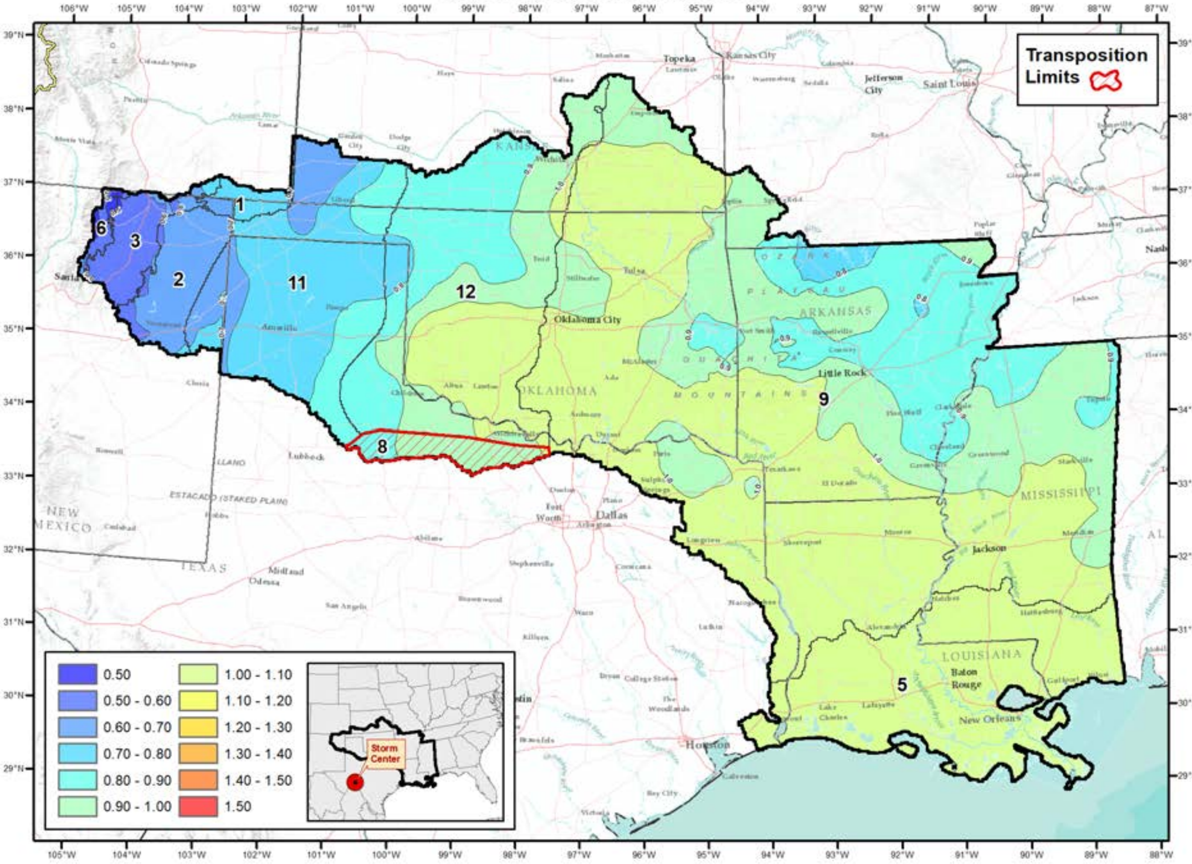
Gridded GTF for Woodward Ranch, TX
May, 1935 (SPAS 1496 DAD zone 1)



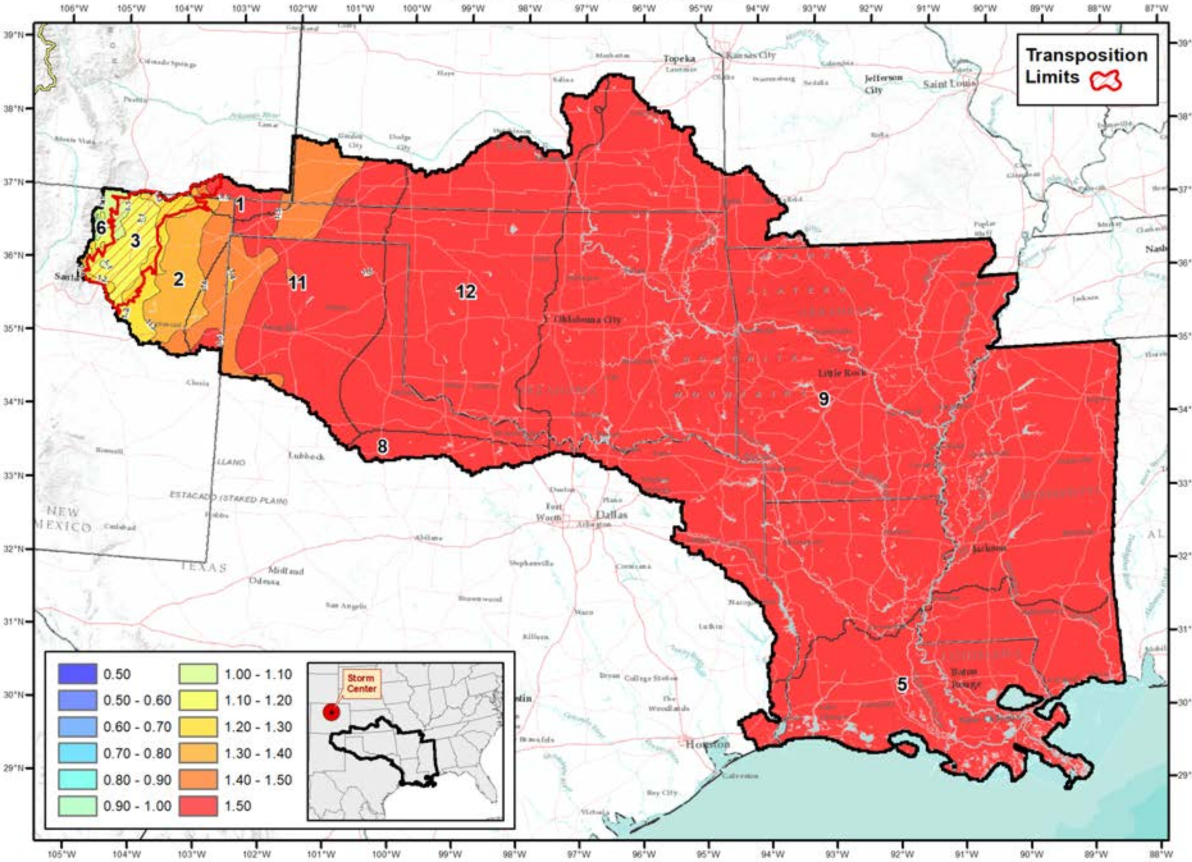
Gridded GTF for Dawson, TX
Oct, 2015 (SPAS 1590 DAD zone 1)



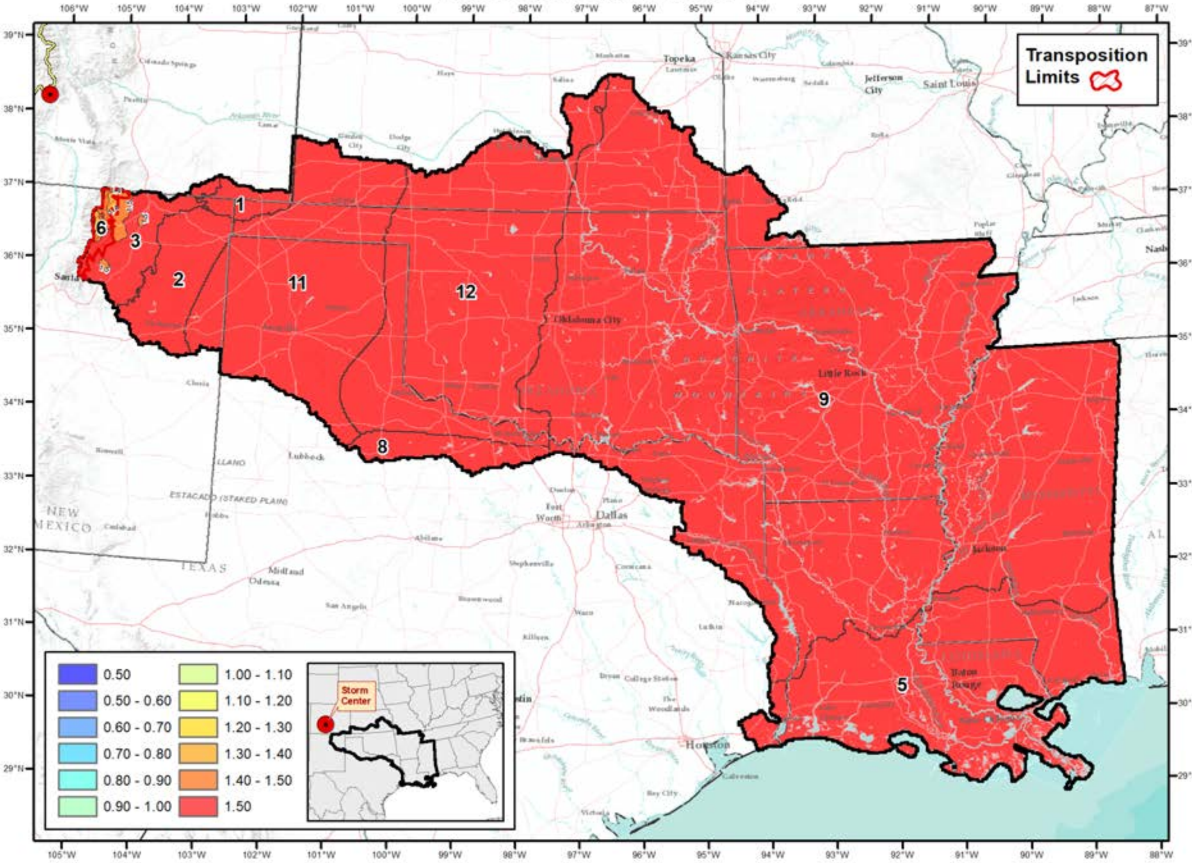
Gridded GTF for Vic Pierce, TX
Jun, 1954 (SPAS 1602 DAD zone 1)



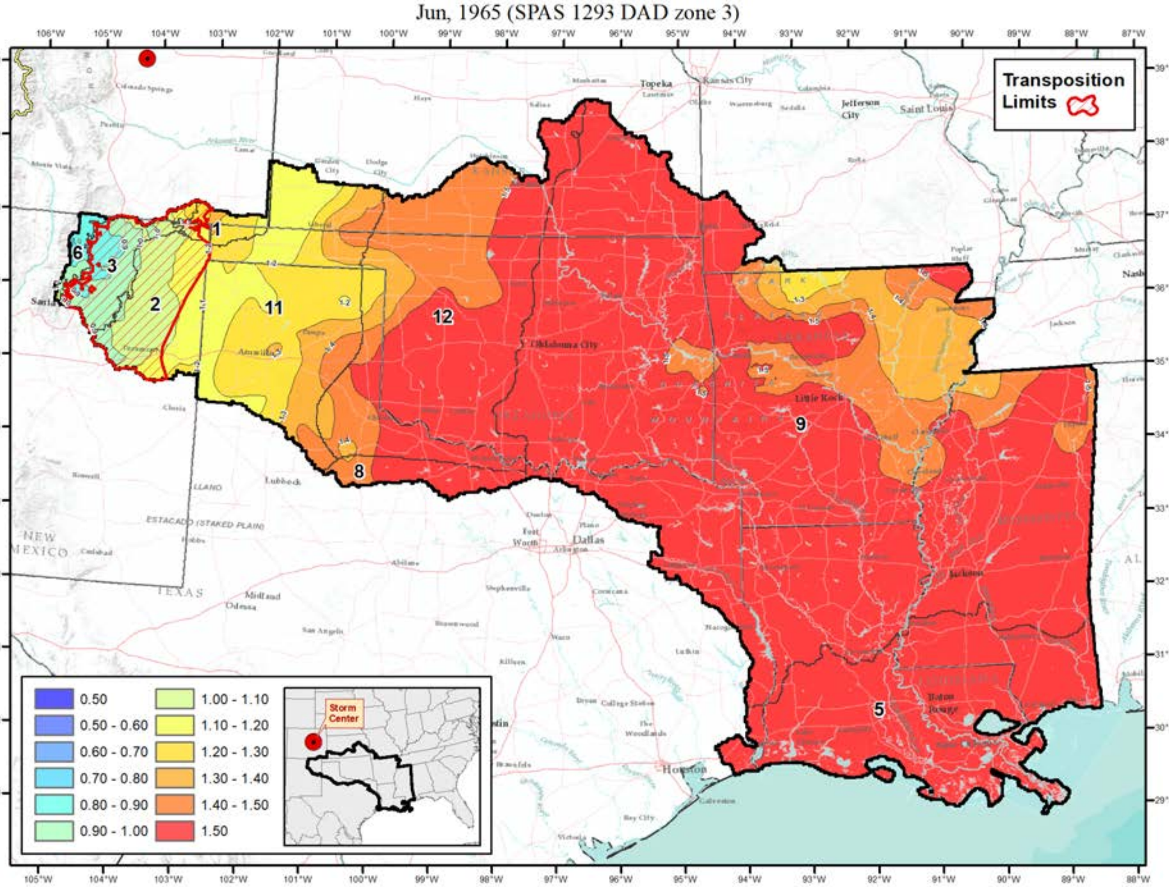
Gridded GTF for Golden, CO
Jun, 1948 (SPAS 1613 DAD zone 1)



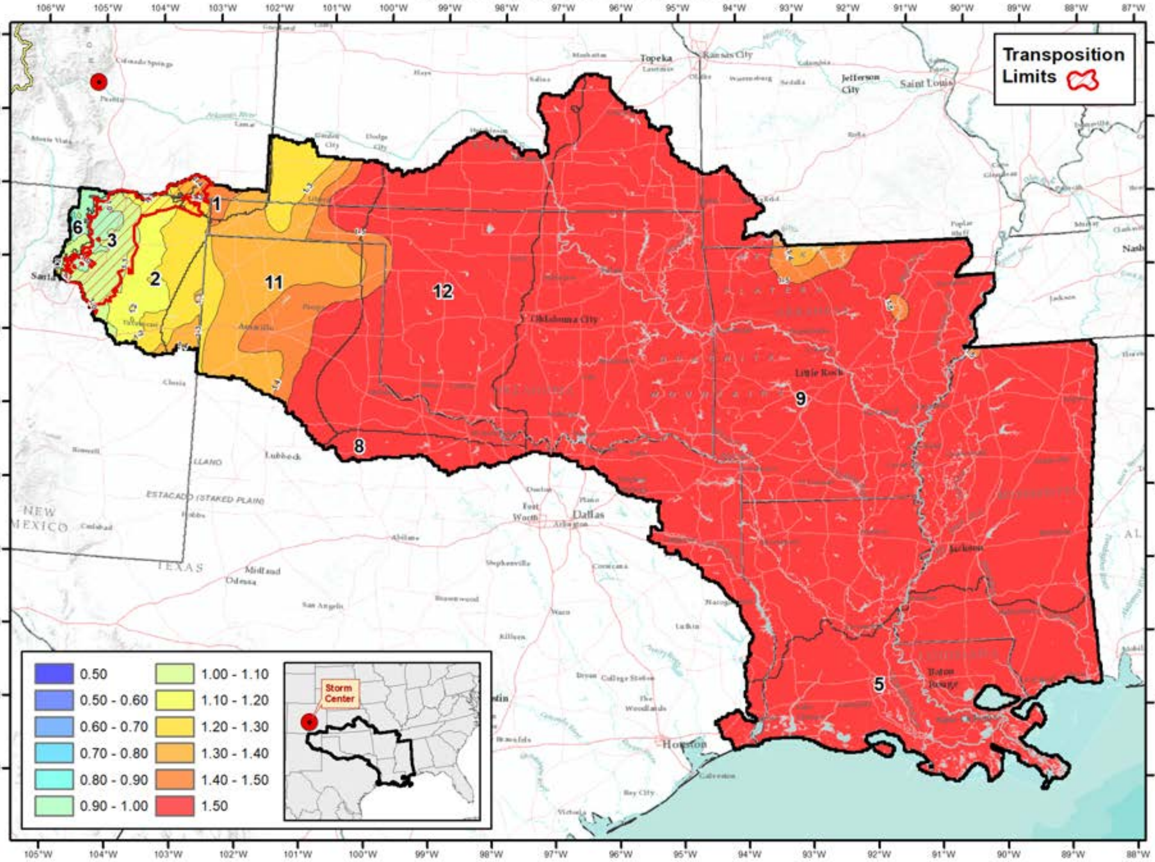
Gridded GTF for Saguache, CO
Jul, 1999 (SPAS 1662 DAD zone 1)



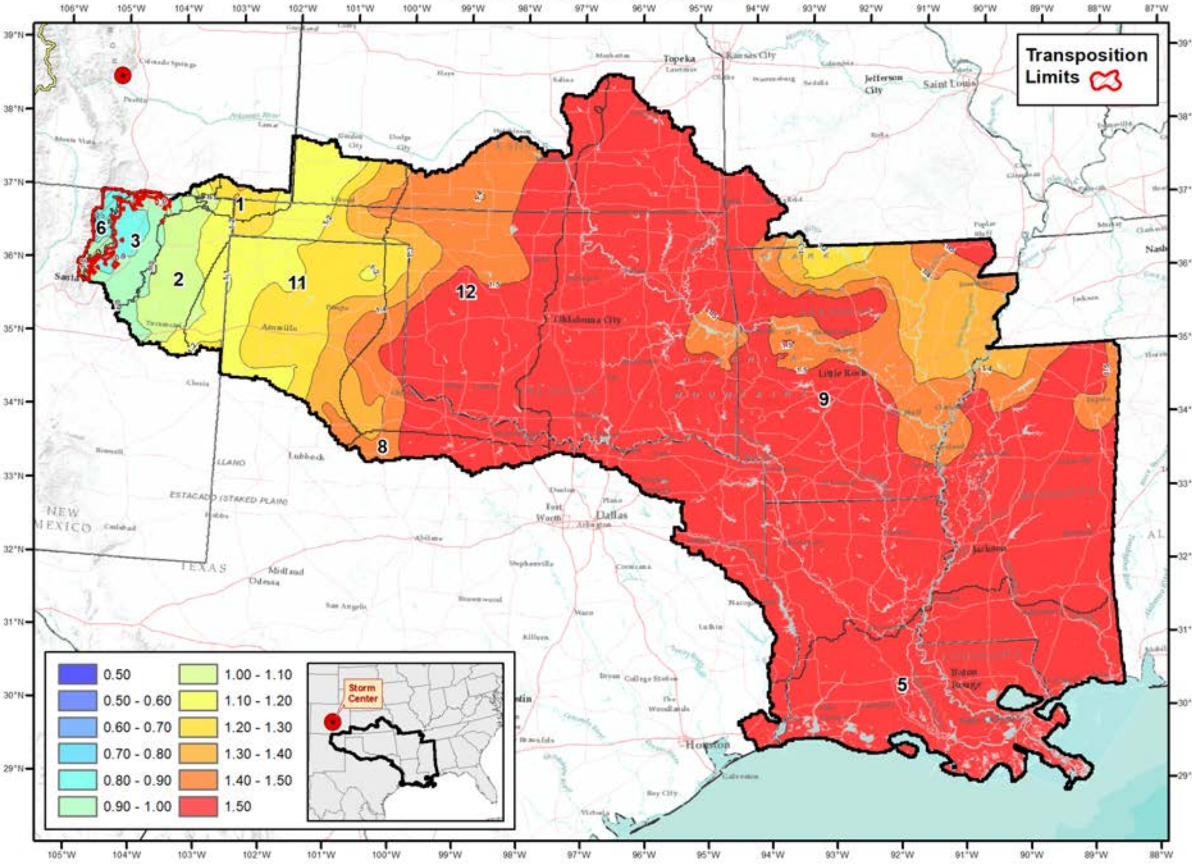
Gridded GTF for Elbert, CO
Jun, 1965 (SPAS 1293 DAD zone 3)



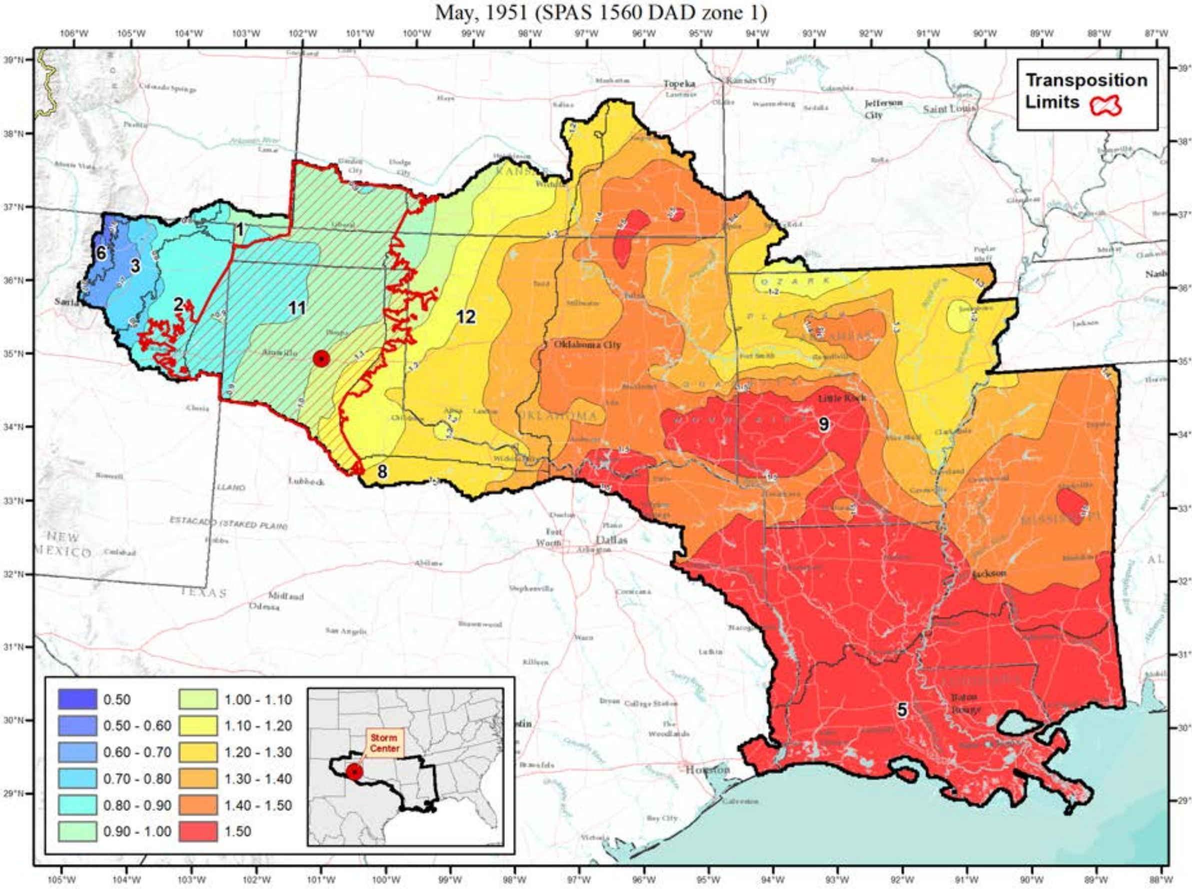
Gridded GTF for Penrose, CO
Jun, 1921 (SPAS 1294 DAD zone 1)



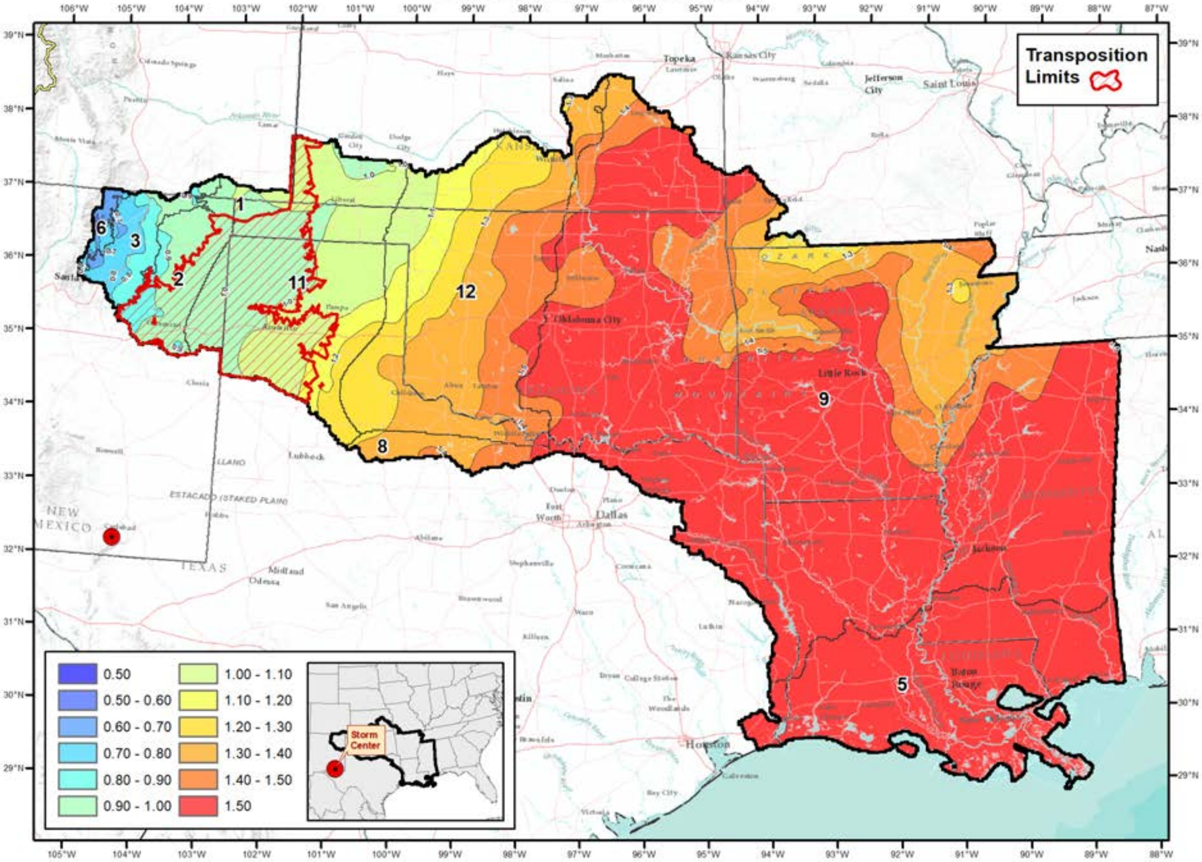
Gridded GTF for Adelaide, CO
Jun, 1921 (SPAS 1294 DAD zone 2)



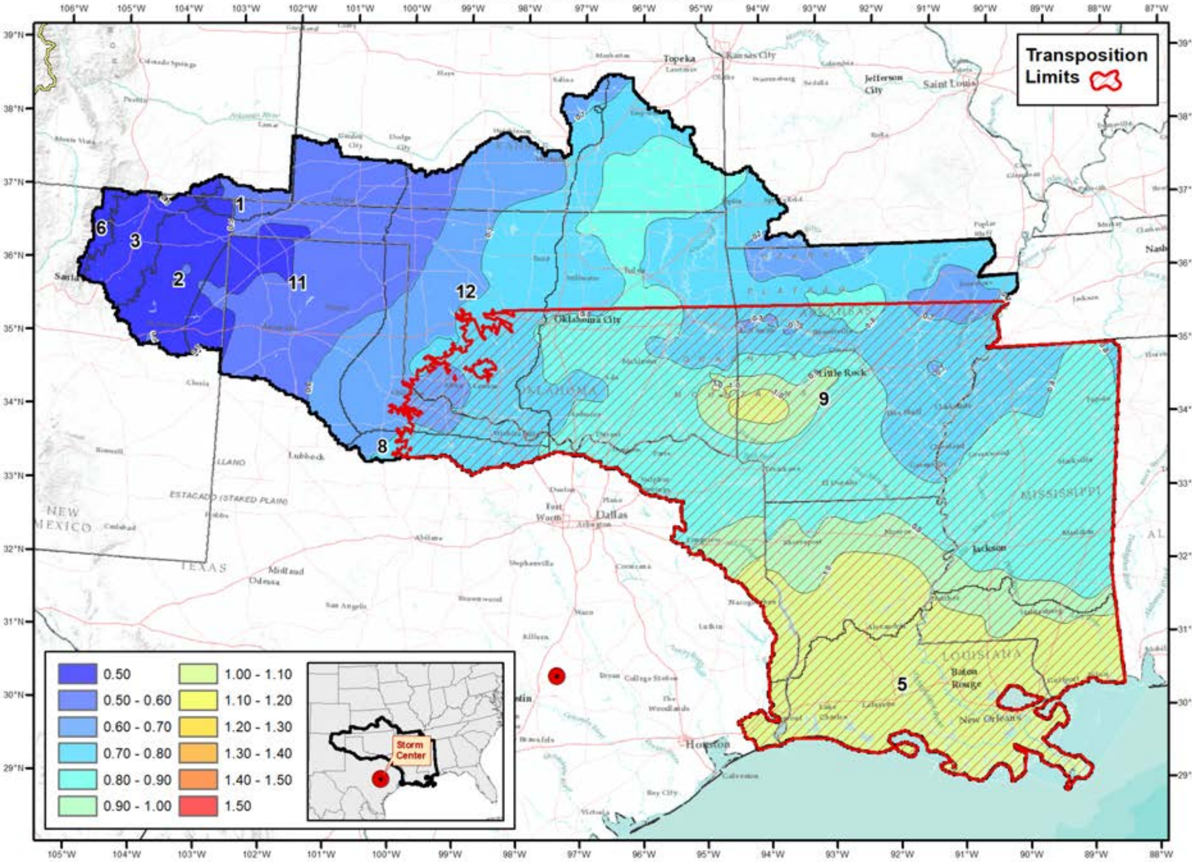
Gridded GTF for Conway, TX
May, 1951 (SPAS 1560 DAD zone 1)



Gridded GTF for Carlsbad, NM
Aug, 1966 (SPAS 1568 DAD zone 1)

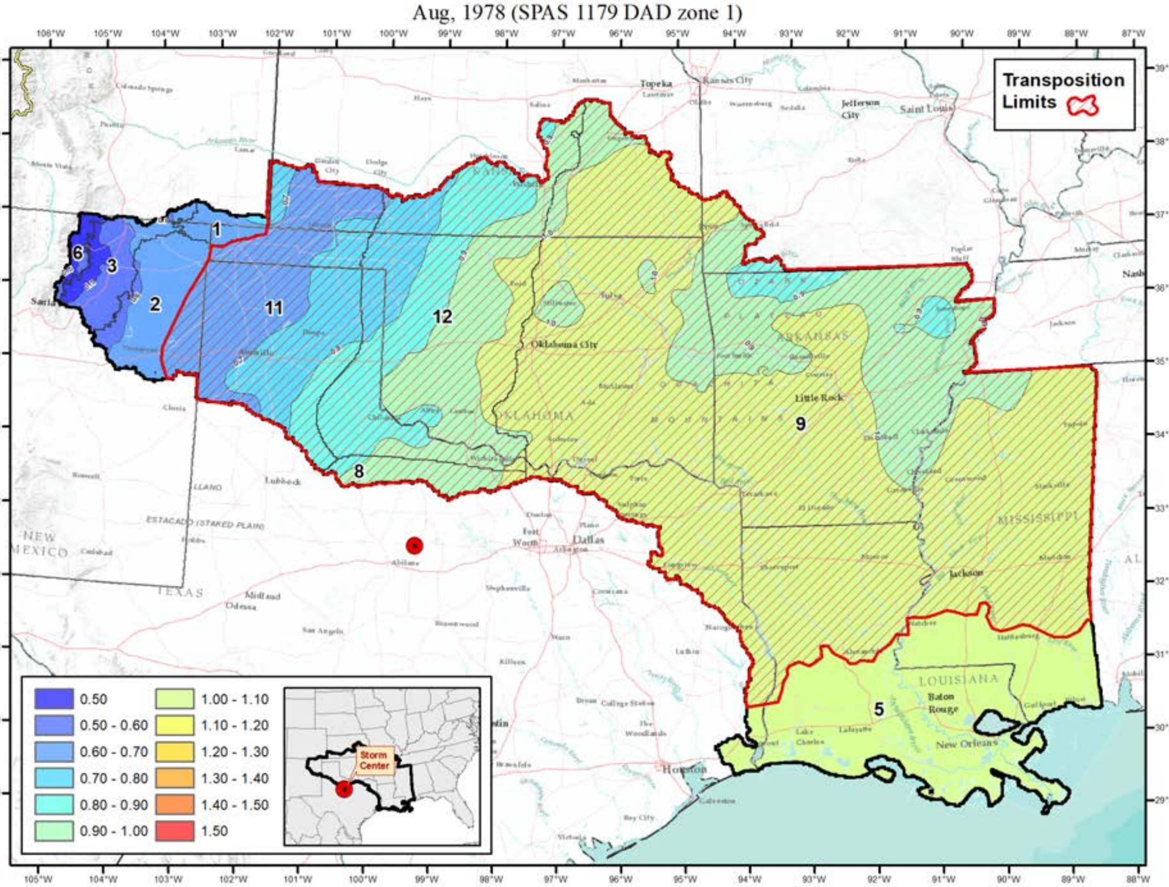


Gridded GTF for Thrall, TX
Sep, 1921 (SPAS 1592 DAD zone 1)

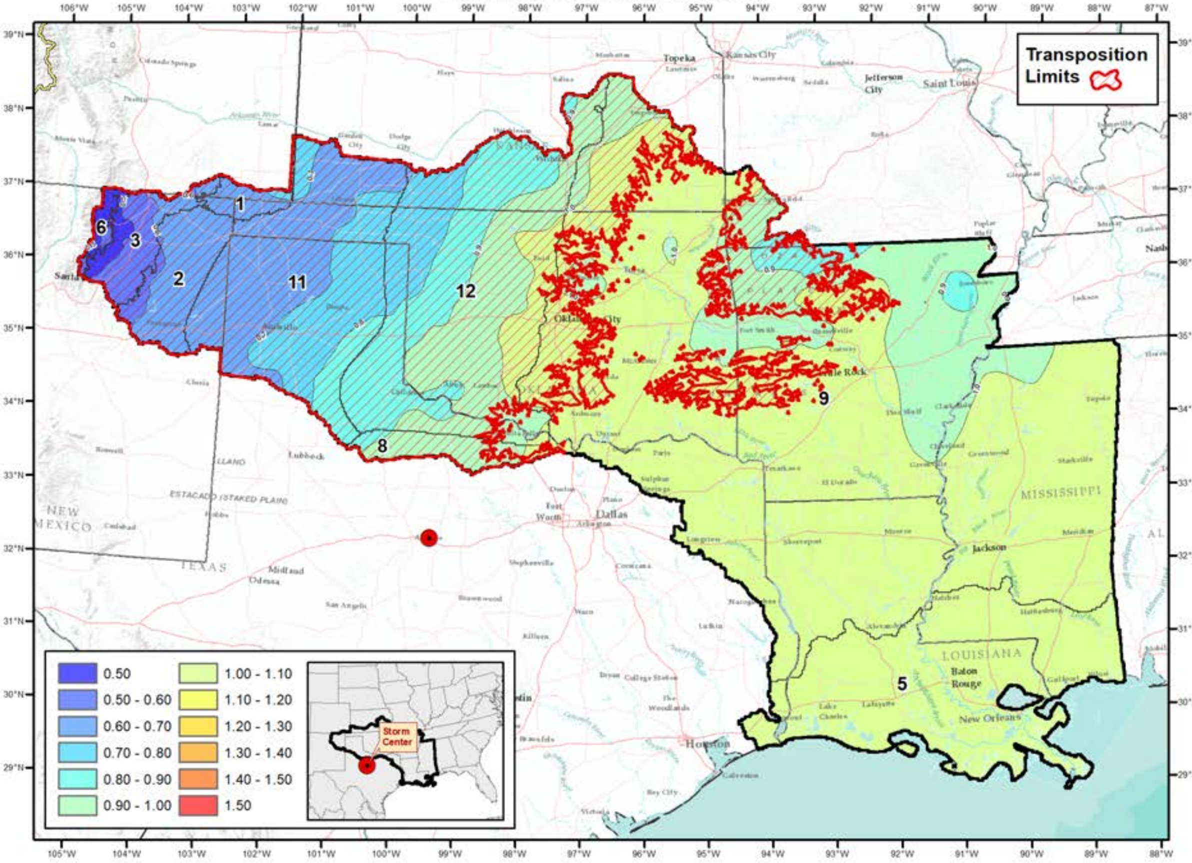


# Tropical Storms

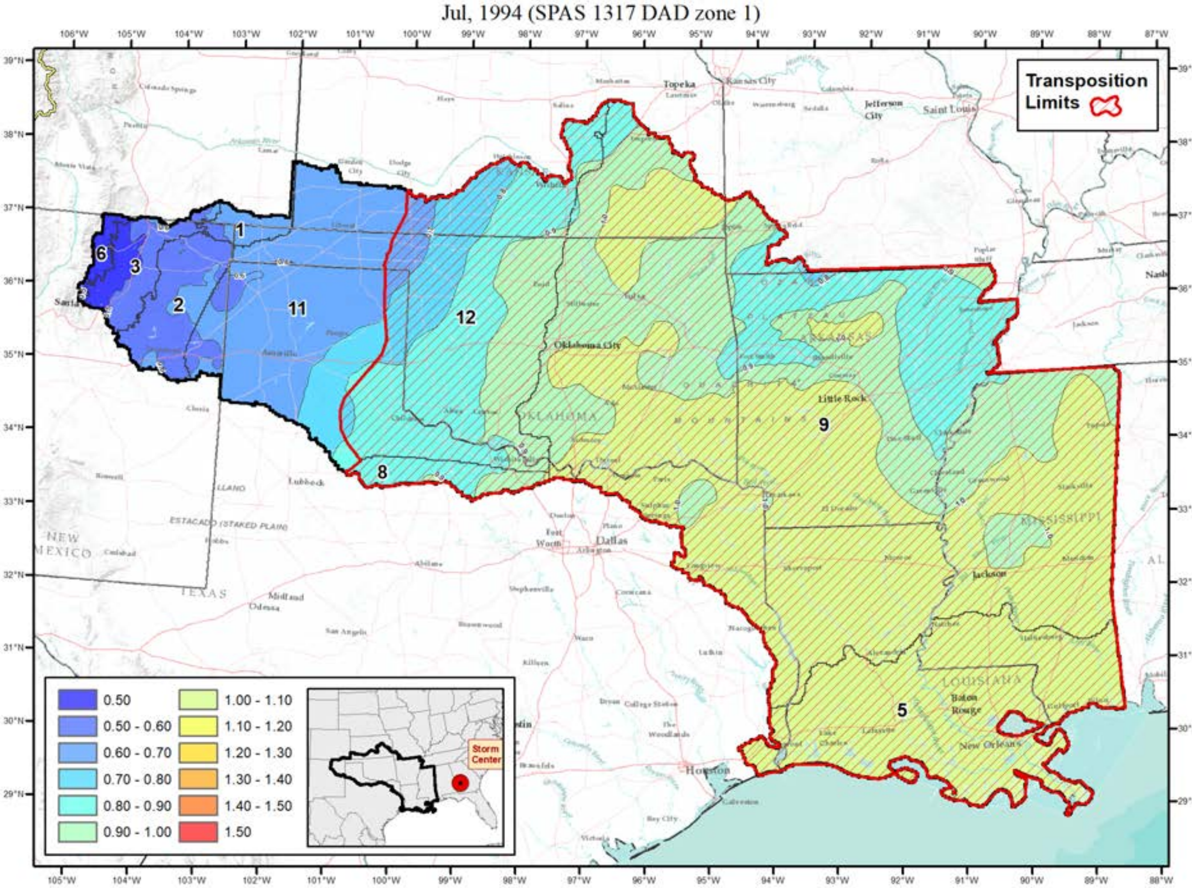
Gridded GTF for Albany, TX
Aug, 1978 (SPAS 1179 DAD zone 1)



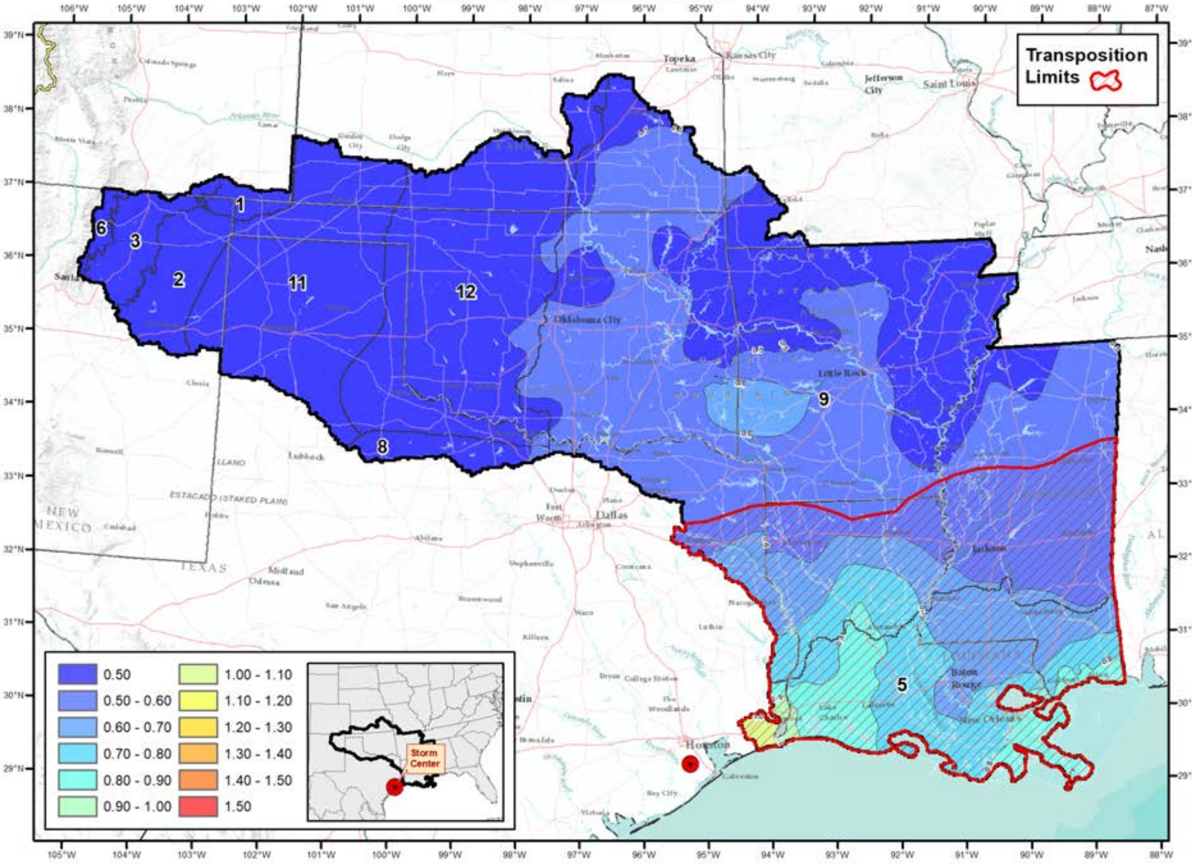
Gridded GTF for Clyde, TX
Oct, 1981 (SPAS 1184 DAD zone 1)



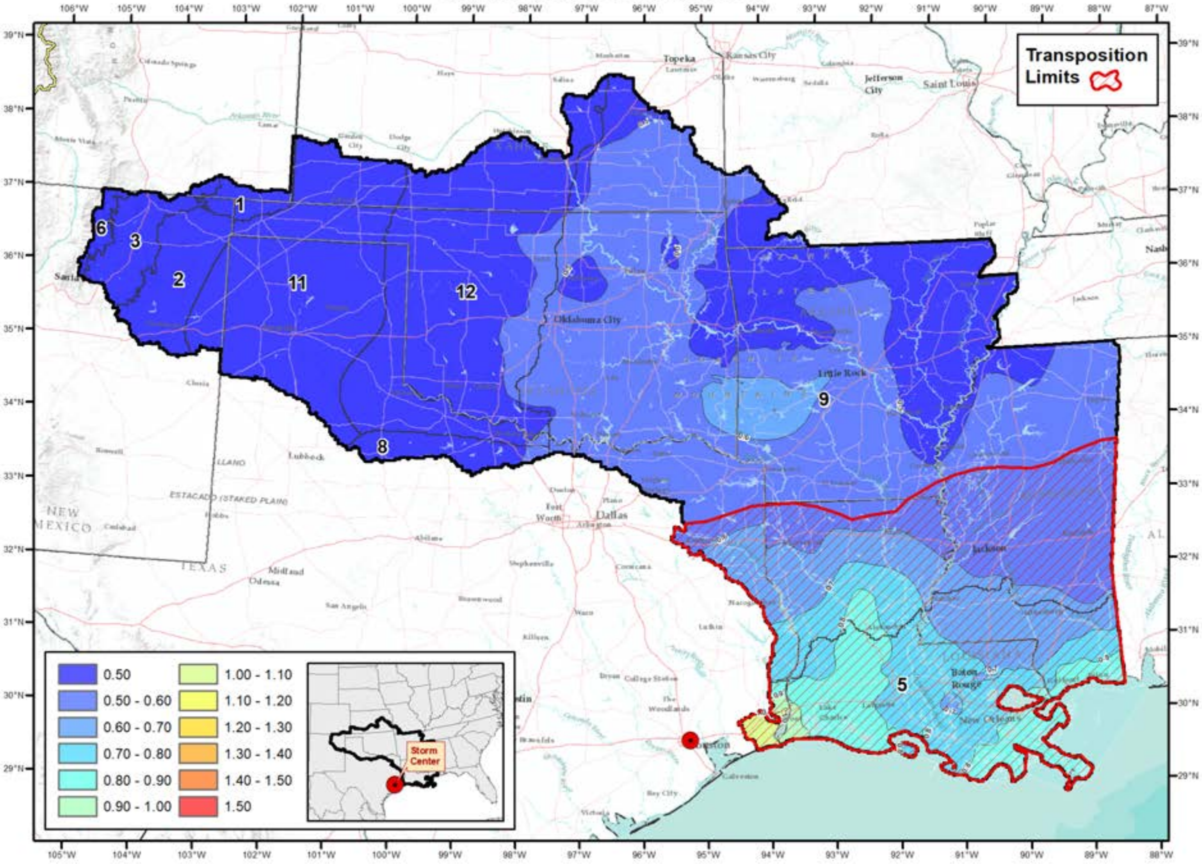
Gridded GTF for Americus, GA
Jul, 1994 (SPAS 1317 DAD zone 1)



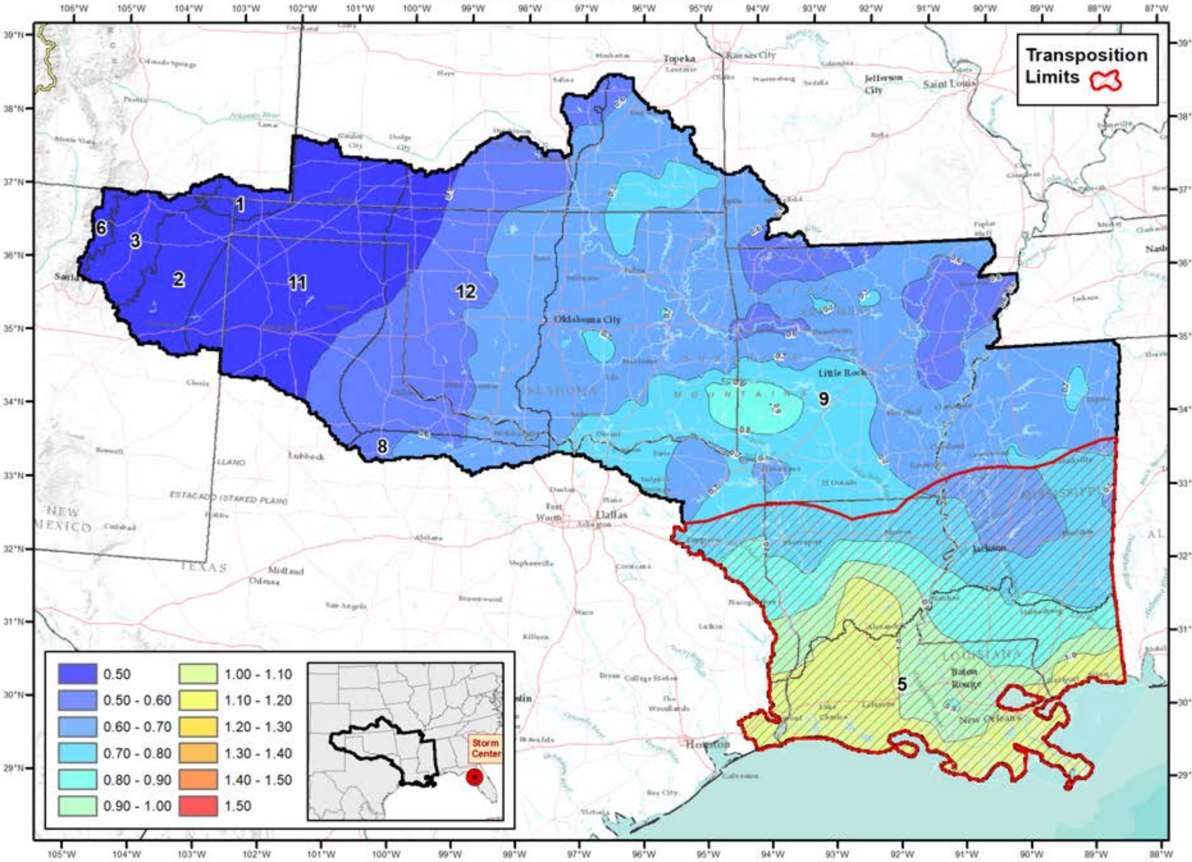
Gridded GTF for Alvin, TX
Jul, 1979 (SPAS 1463 DAD zone 1)



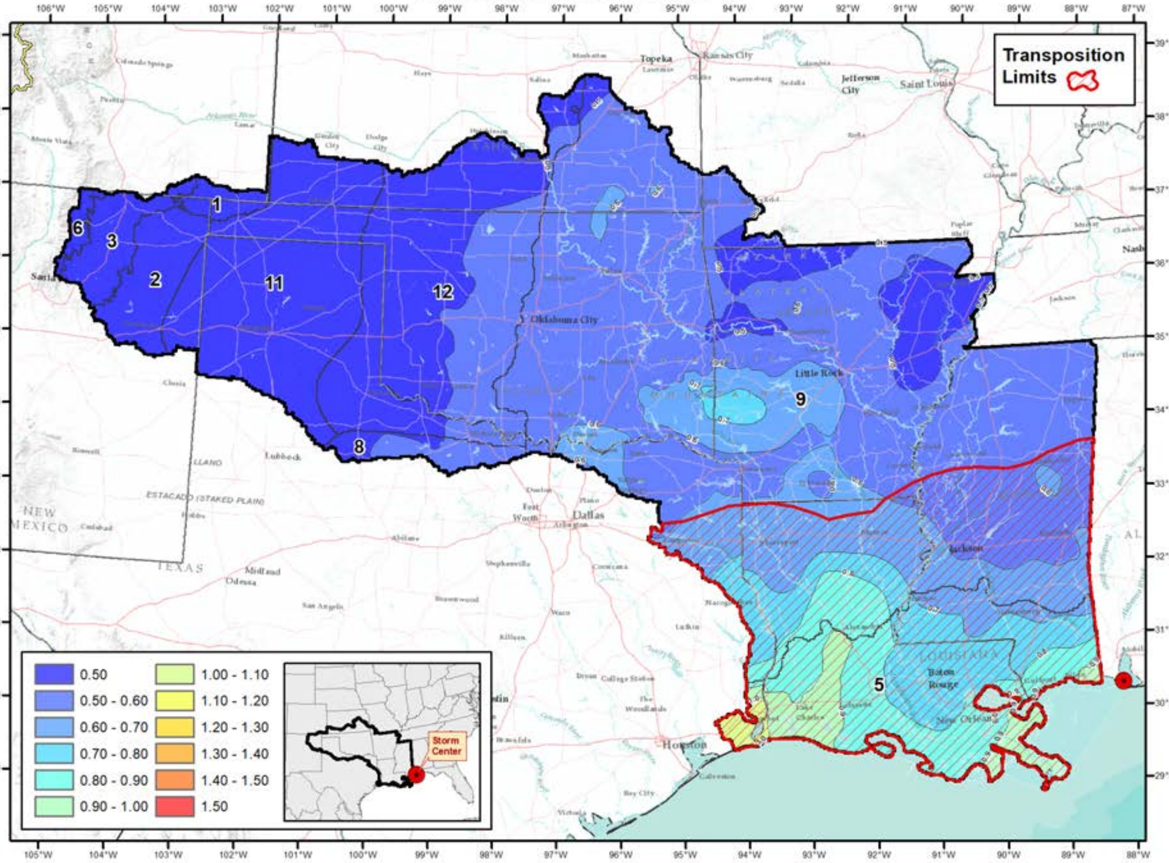
Gridded GTF for Houston, TX
Jun, 2001 (SPAS 1464 DAD zone 1)



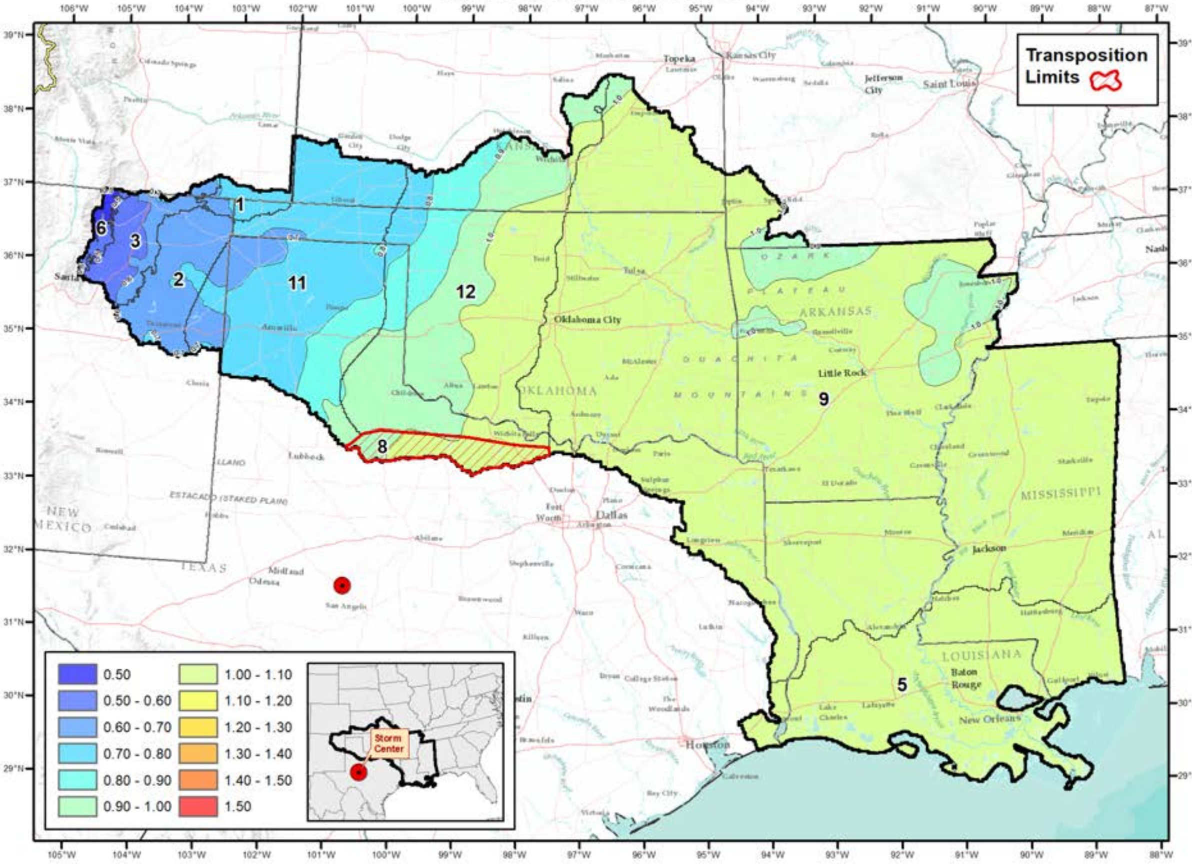
Gridded GTF for Yankeetown, FL
Sep, 1950 (SPAS 1519 DAD zone 1)



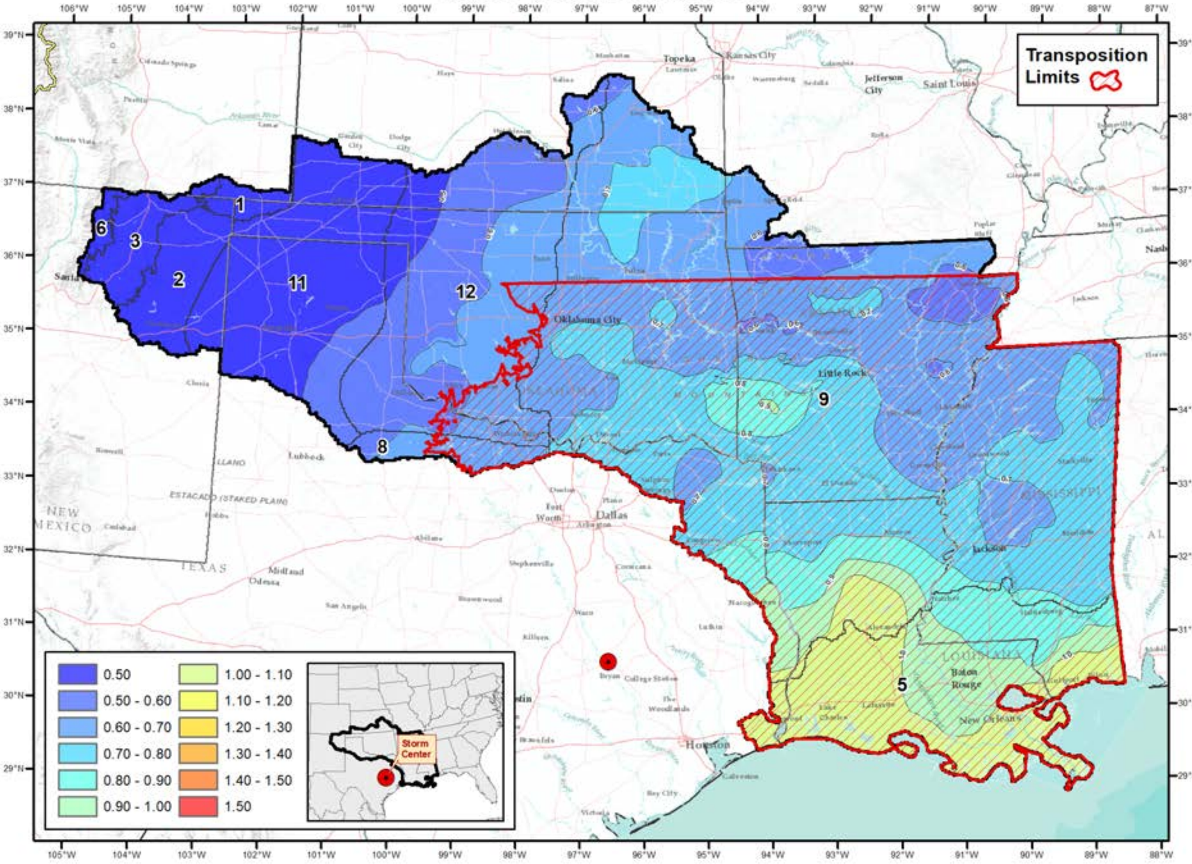
Gridded GTF for Dauphin Island, AL
Jul, 1997 (SPAS 1569 DAD zone 1)



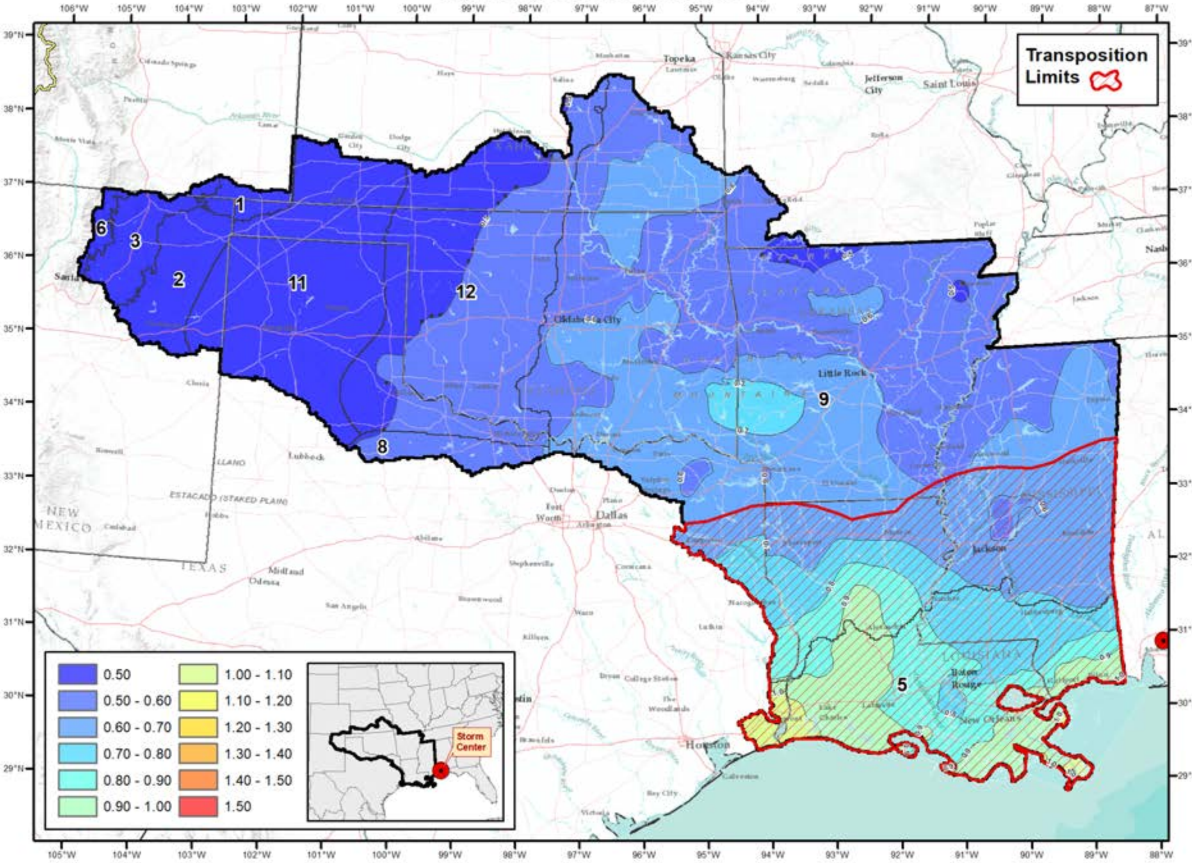
Gridded GTF for Broome, TX
Sep, 1936 (SPAS 1582 DAD zone 1)



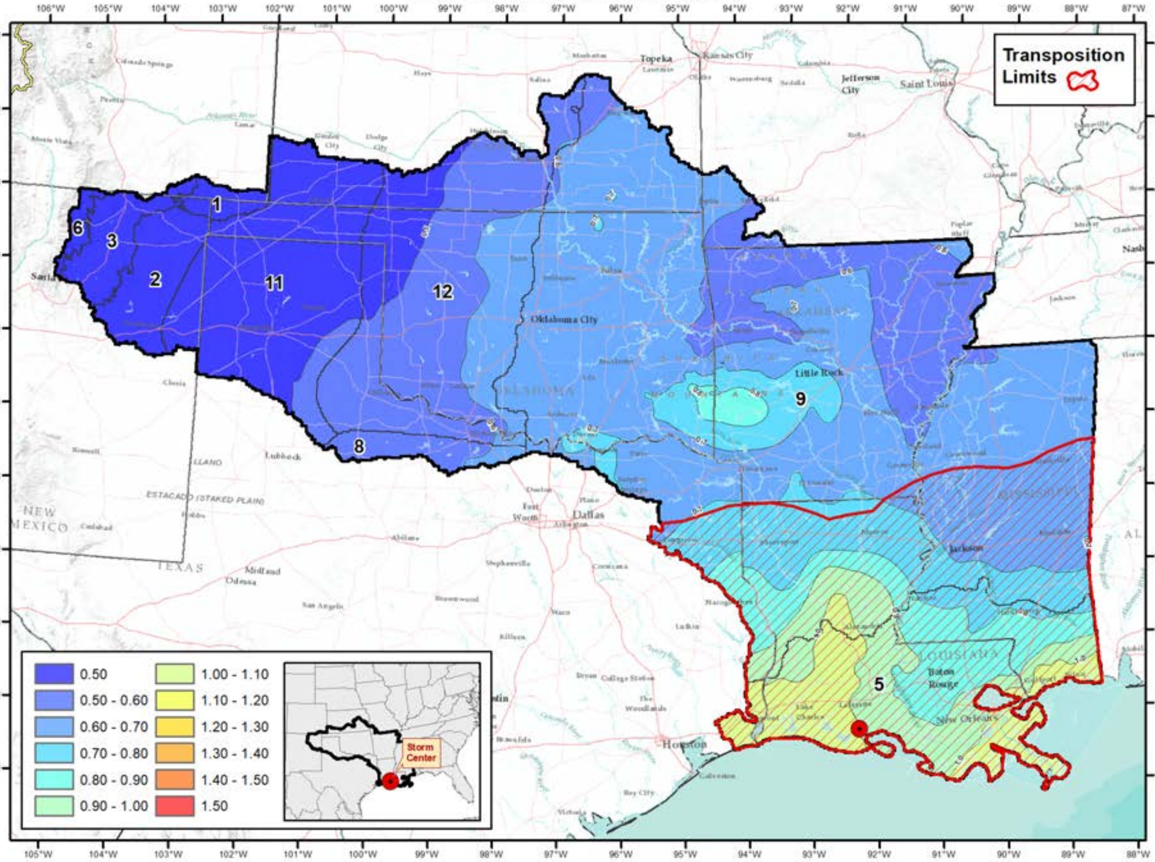
Gridded GTF for Hearne, TX
Jun, 1899 (SPAS 1591 DAD zone 1)



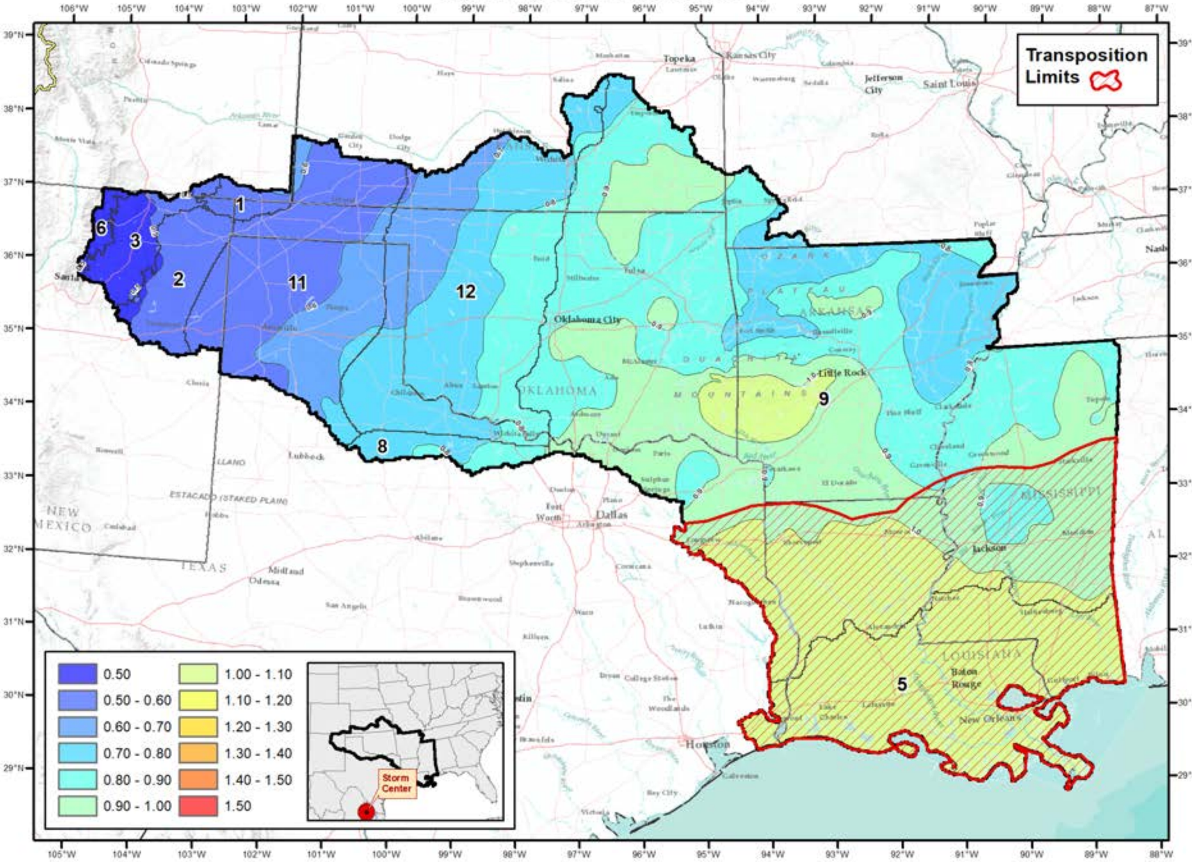
Gridded GTF for Munson, FL
Sep, 1998 (SPAS 1593 DAD zone 1)



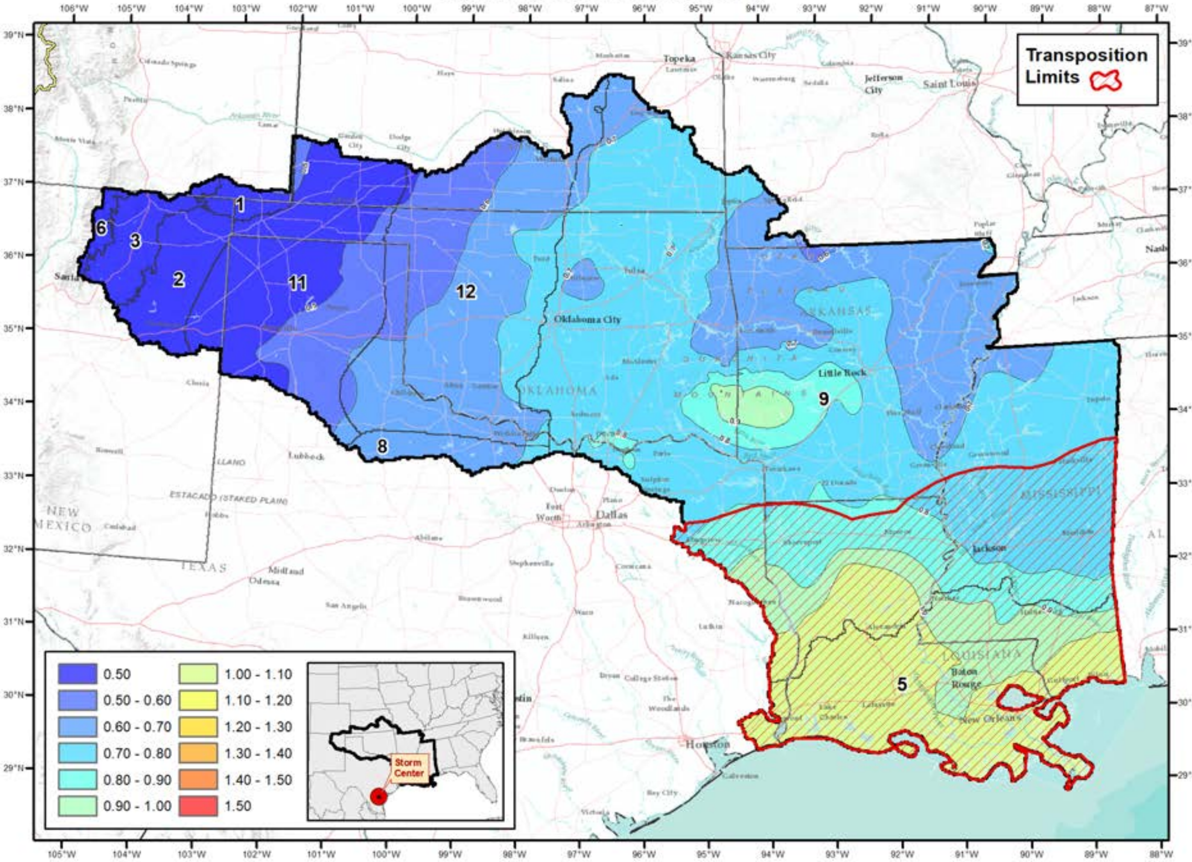
Gridded GTF for Miller Island, LA
Aug, 1940 (SPAS 1596 DAD zone 1)



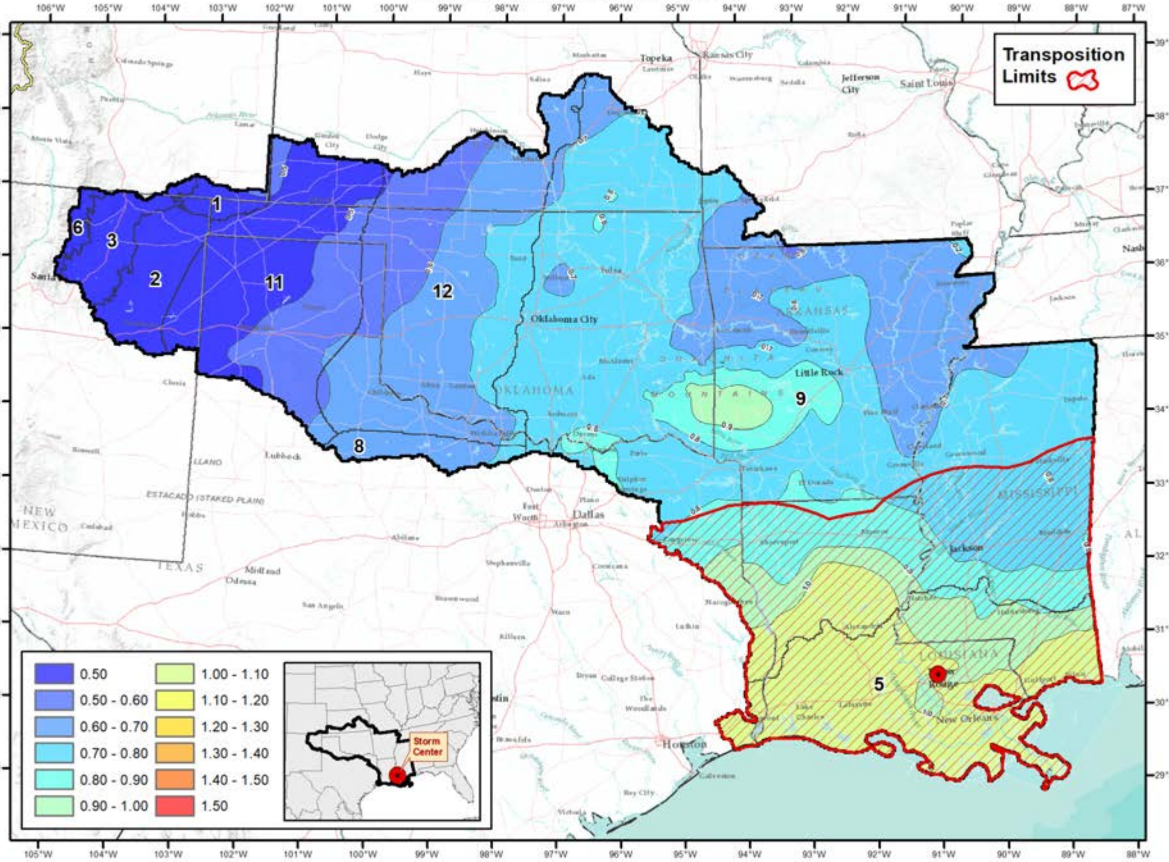
Gridded GTF for Sombreretillo, MX
Sep, 1967 (SPAS 1601 DAD zone 1)



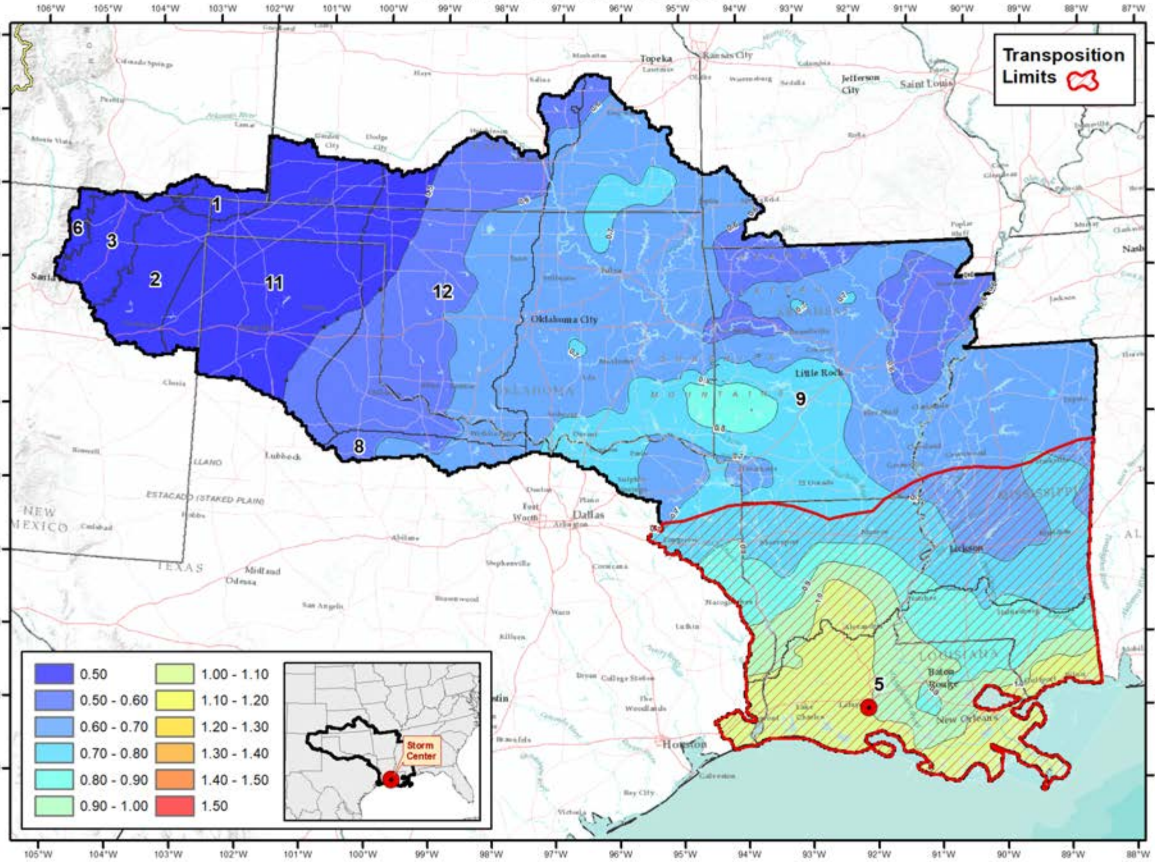
Gridded GTF for Dinero, MX
Sep, 1967 (SPAS 1601 DAD zone 2)



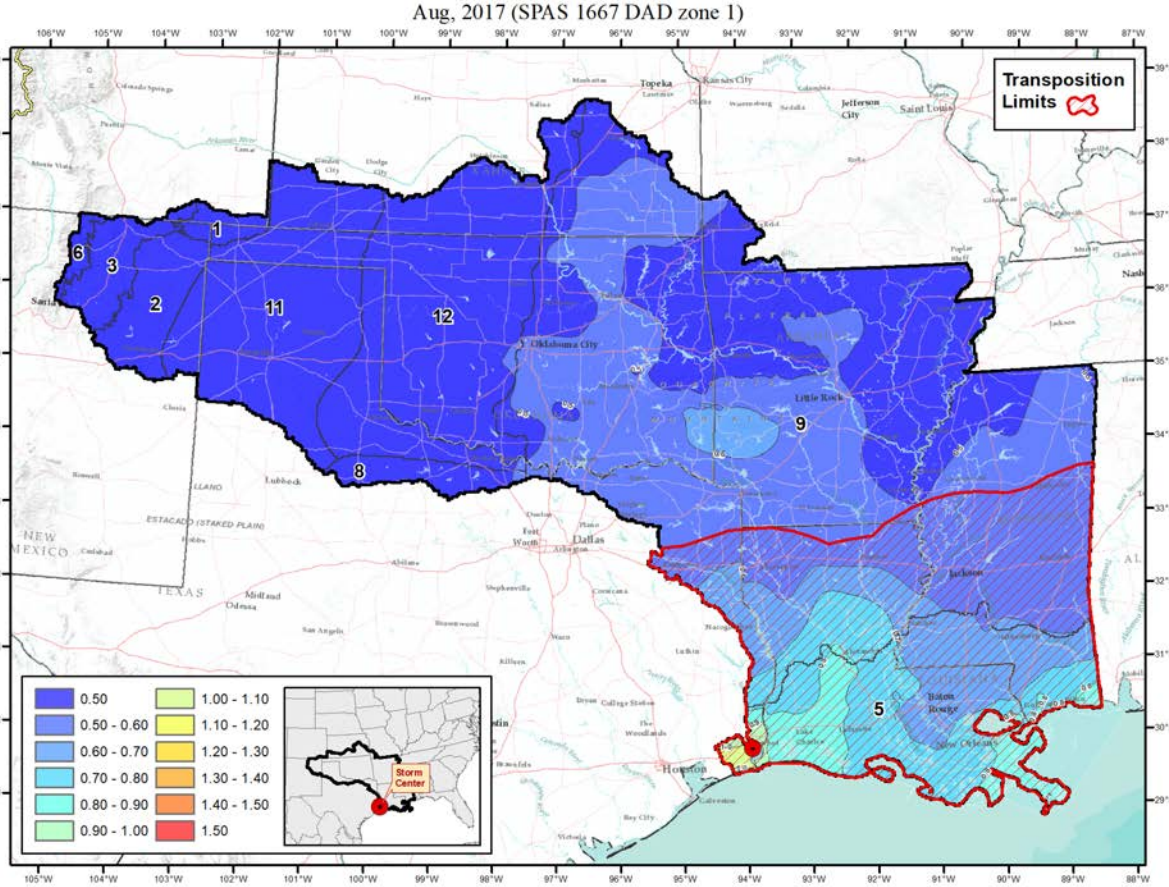
Gridded GTF for Watson, LA
Aug, 2016 (SPAS 1631 DAD zone 1)



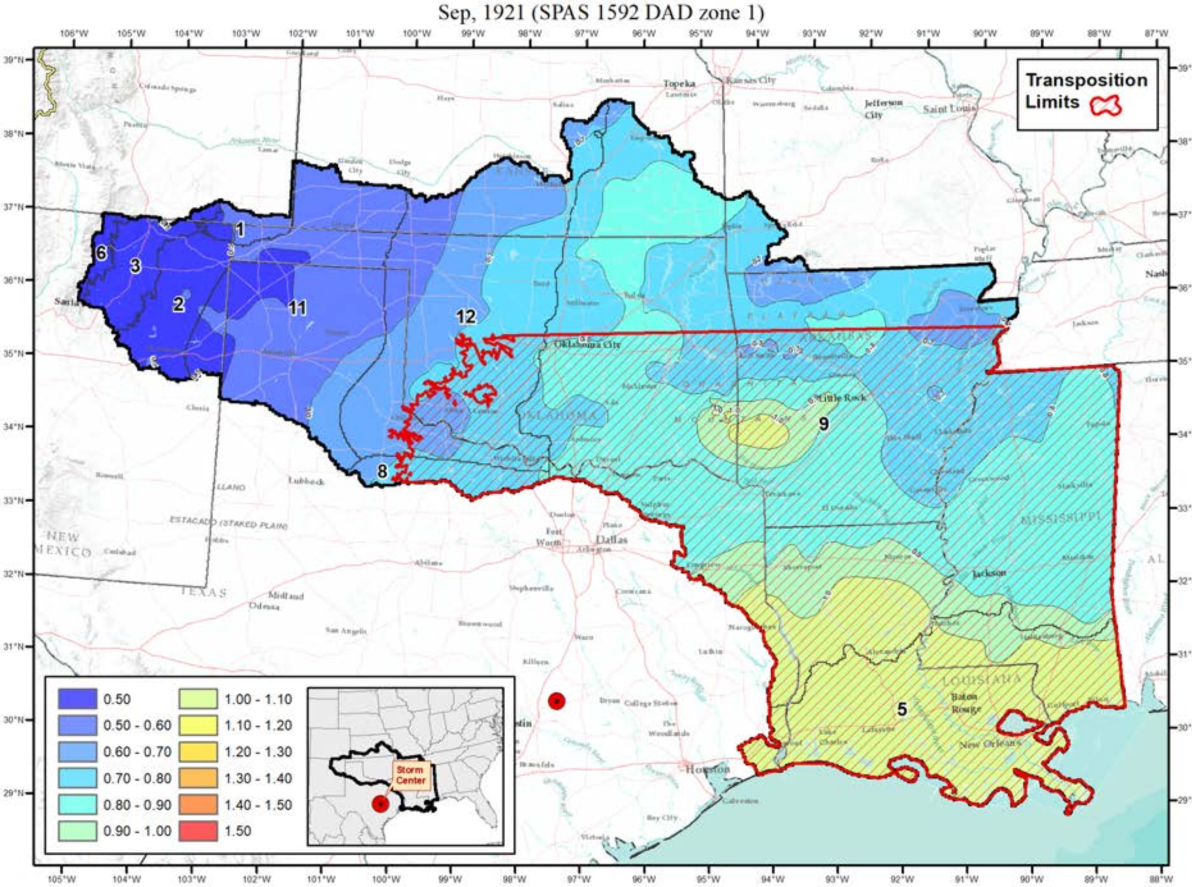
Gridded GTF for Lafayette, LA
Aug, 2016 (SPAS 1631 DAD zone 2)



Gridded GTF for Harvey, TX
Aug, 2017 (SPAS 1667 DAD zone 1)



Gridded GTF for Thrall, TX
Sep, 1921 (SPAS 1592 DAD zone 1)

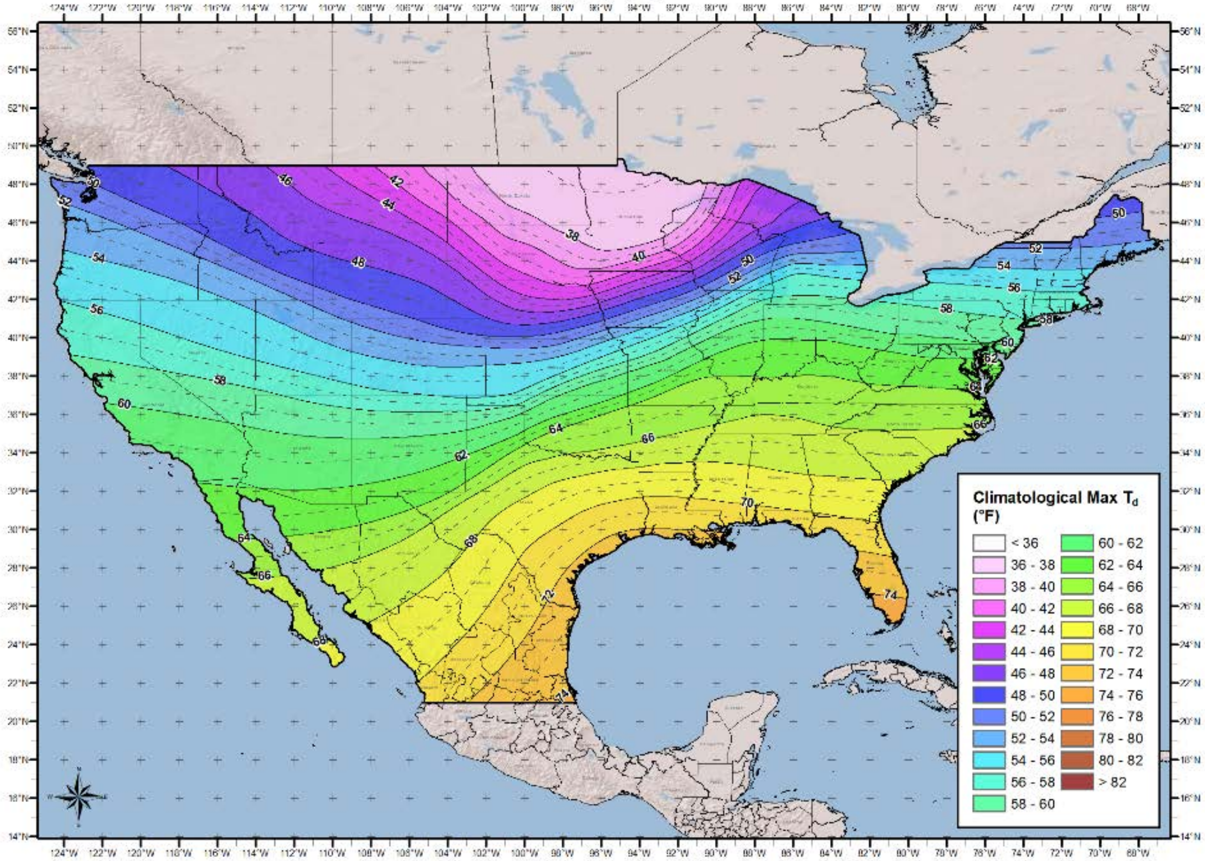


# Appendix C

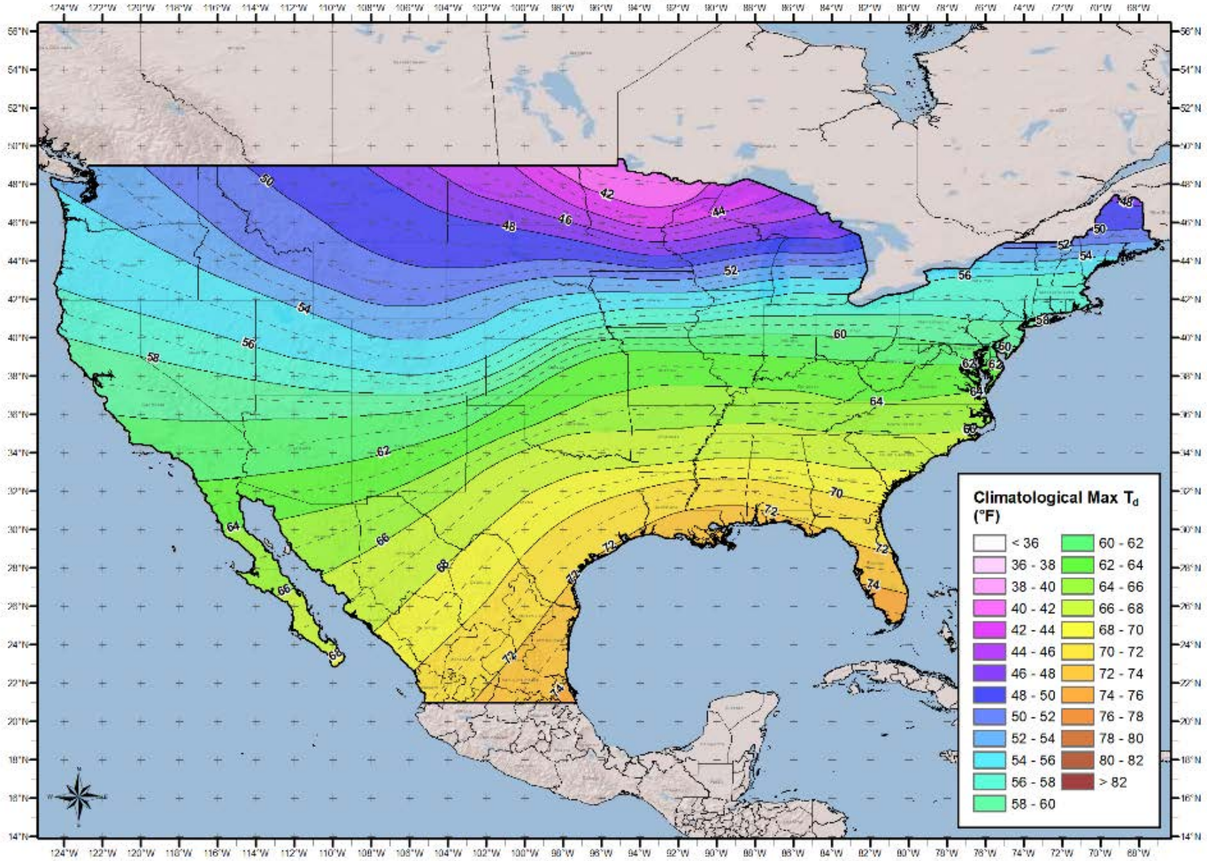
## 100-year Return Frequency Maximum Average Dew Point Temperature Climatology Maps

# 6-hour 1000mb Dew Point Maps

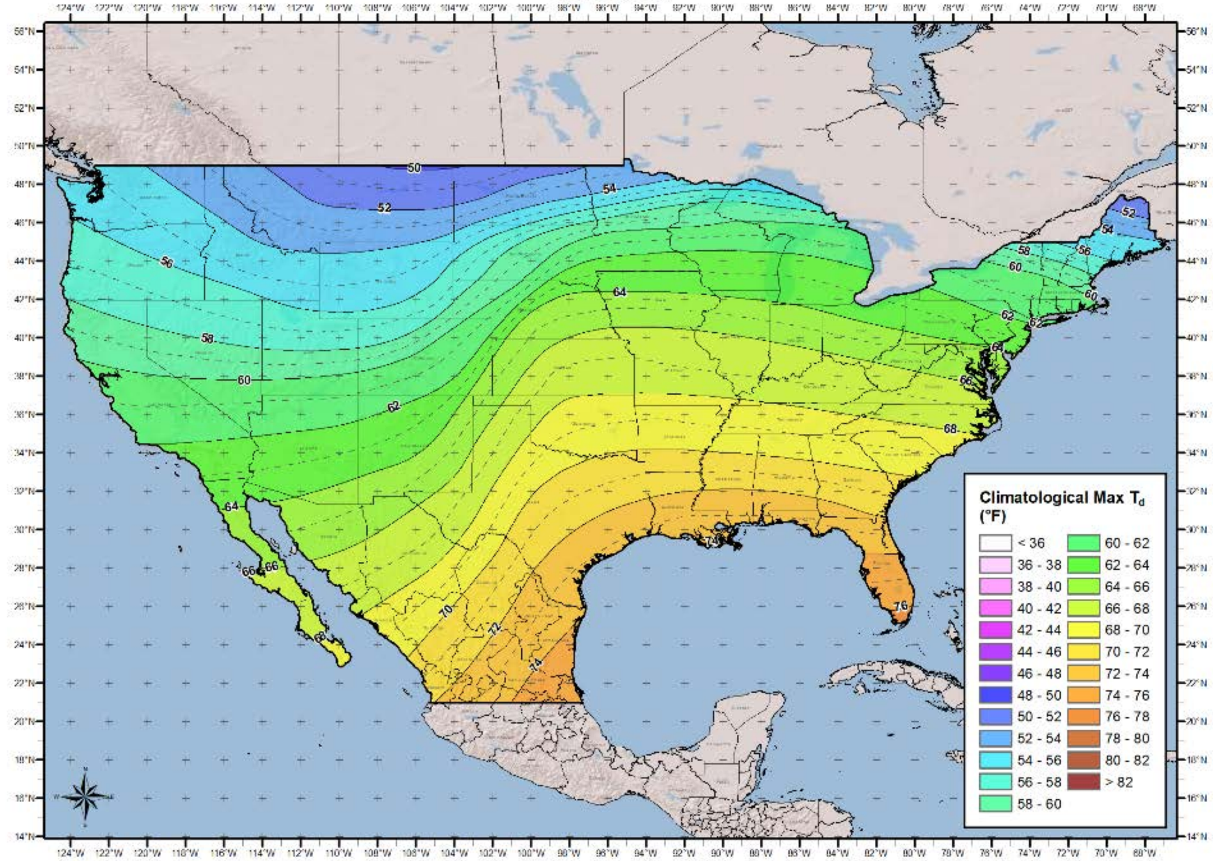
## 100-year Return Frequency 6-hour Maximum Dew Point Climatology
January (°F)



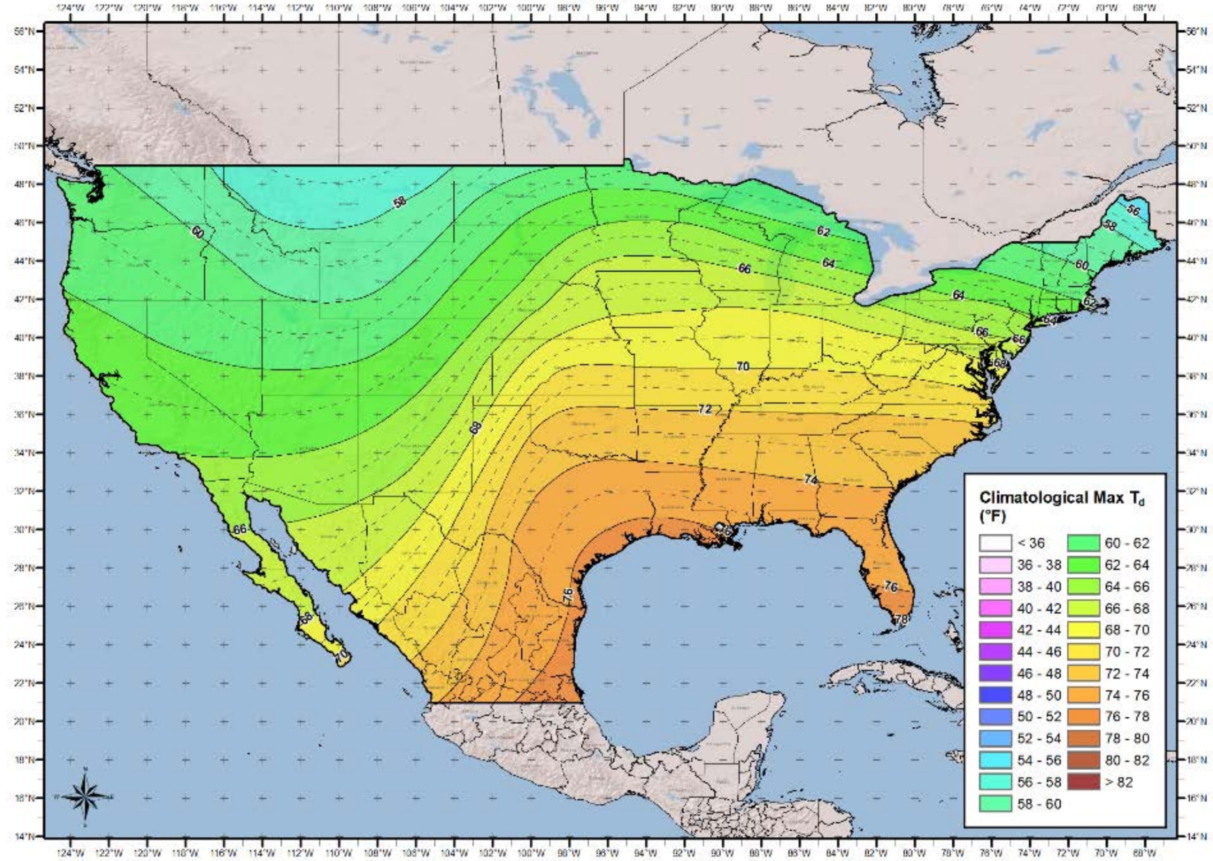
100-year Return Frequency 6-hour Maximum Dew Point Climatology
February (°F)



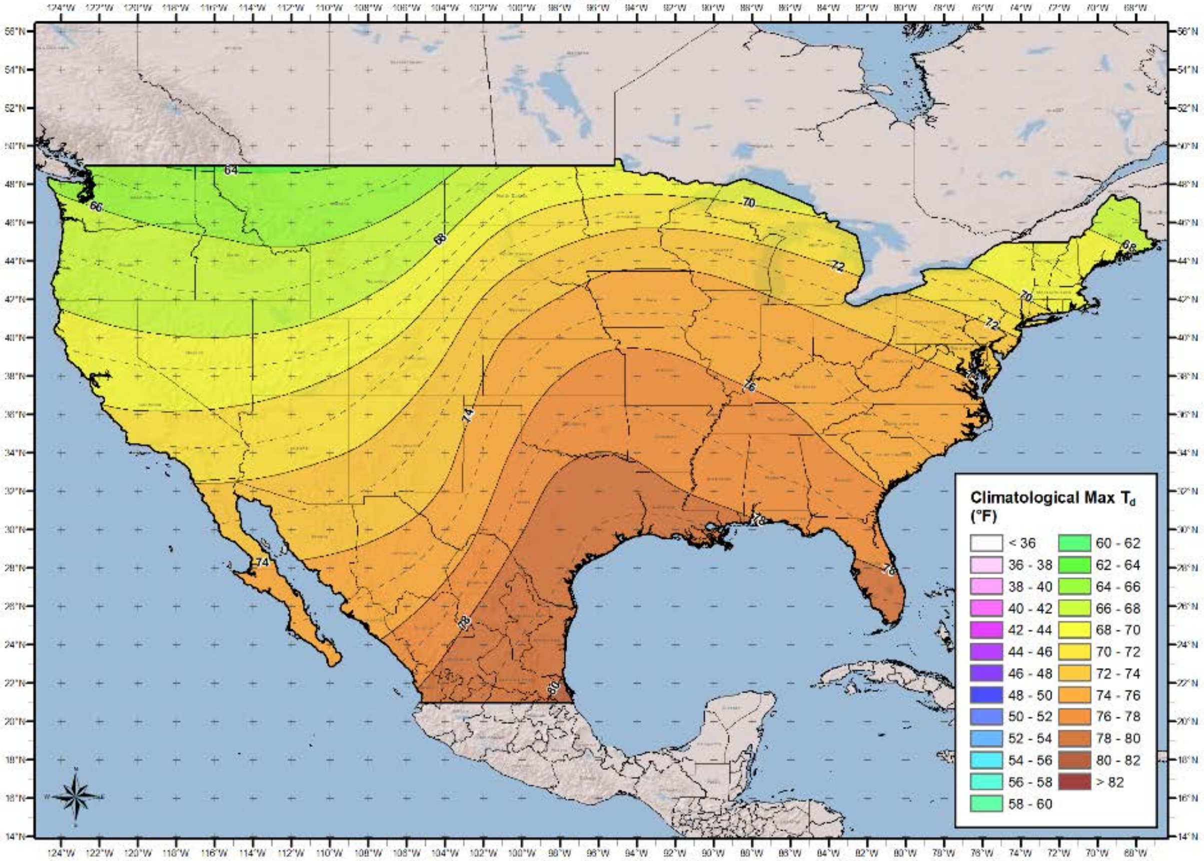
100-year Return Frequency 6-hour Maximum Dew Point Climatology
March (°F)



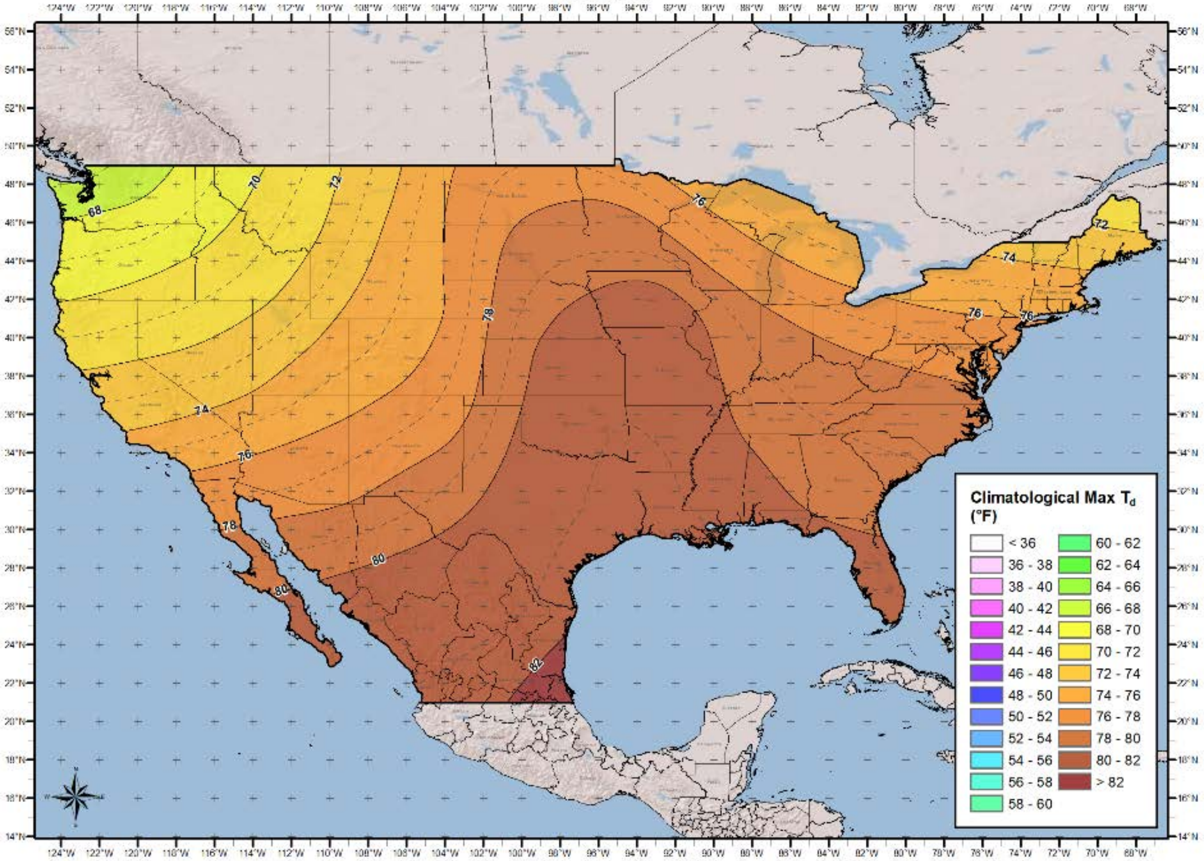
100-year Return Frequency 6-hour Maximum Dew Point Climatology
April (°F)



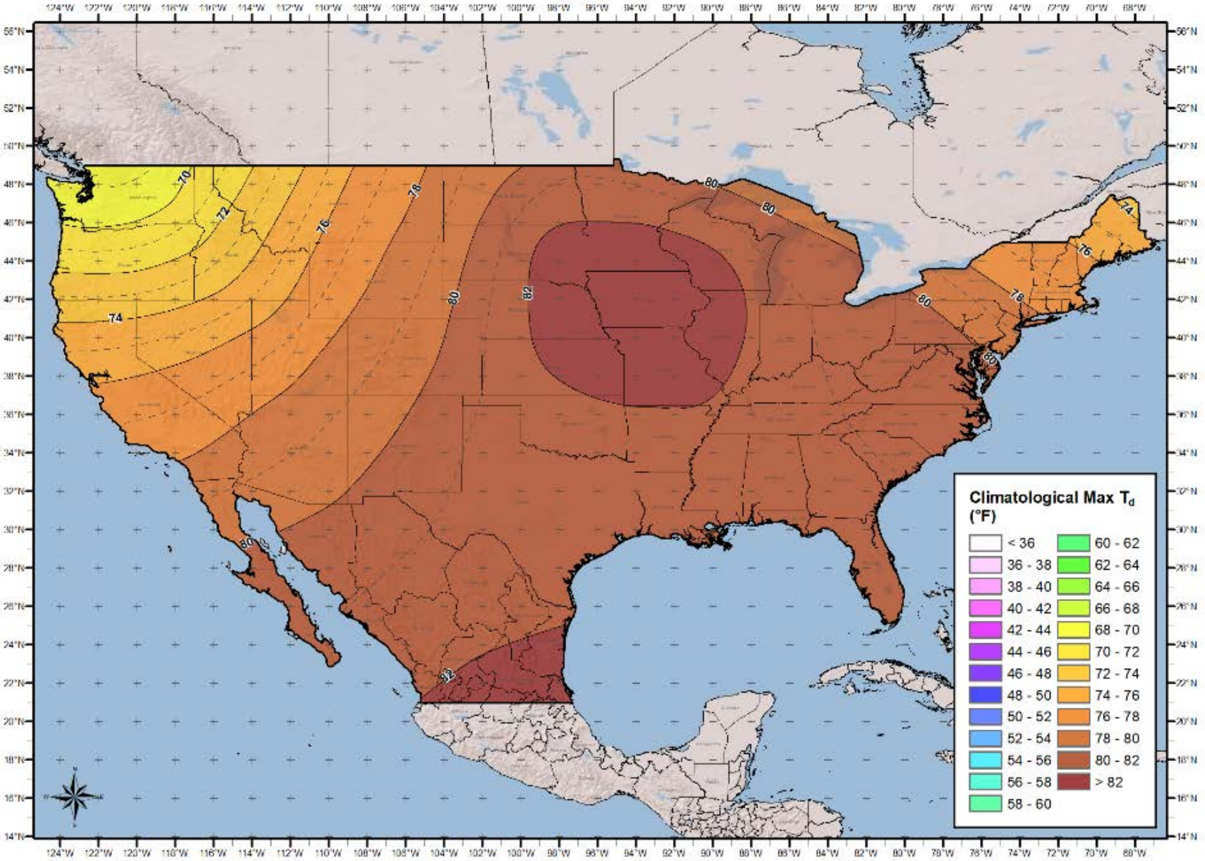
## 100-year Return Frequency 6-hour Maximum Dew Point Climatology
## May (°F)



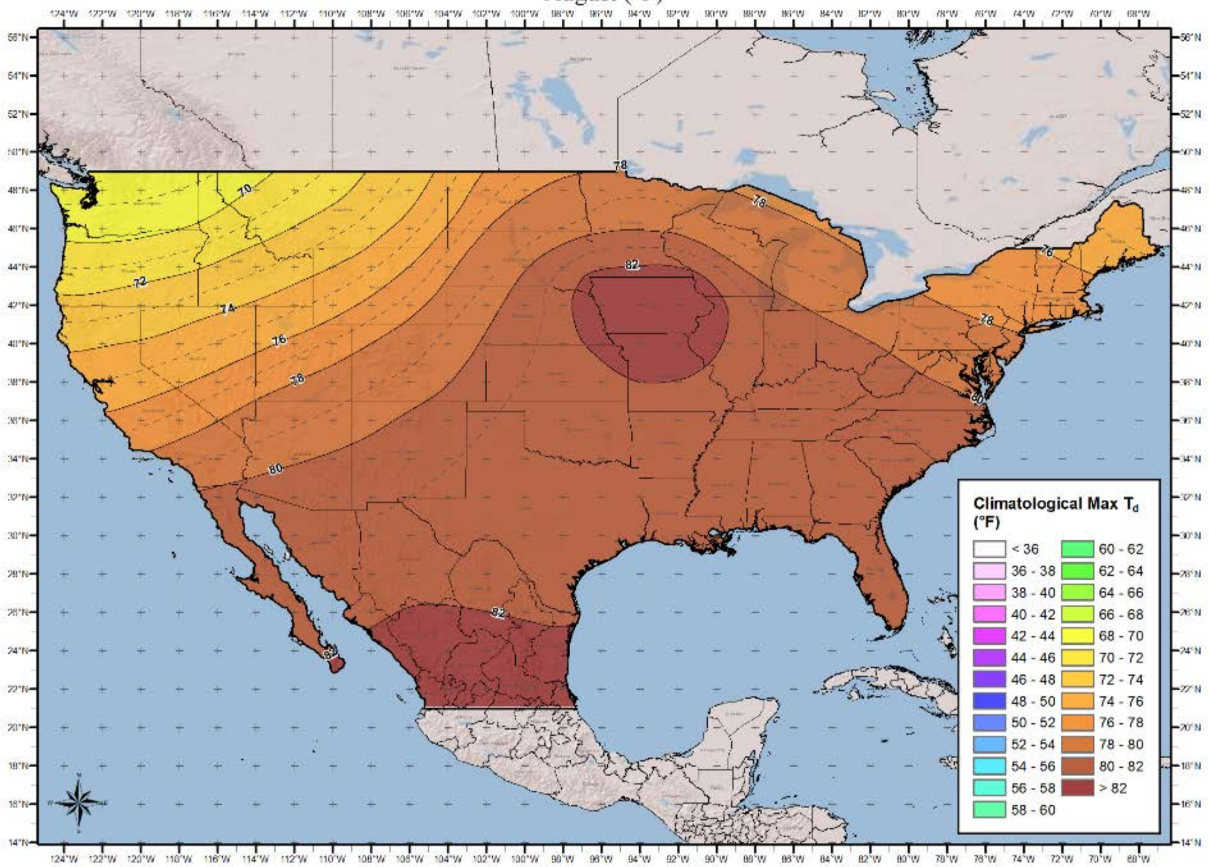
## 100-year Return Frequency 6-hour Maximum Dew Point Climatology
## June (°F)



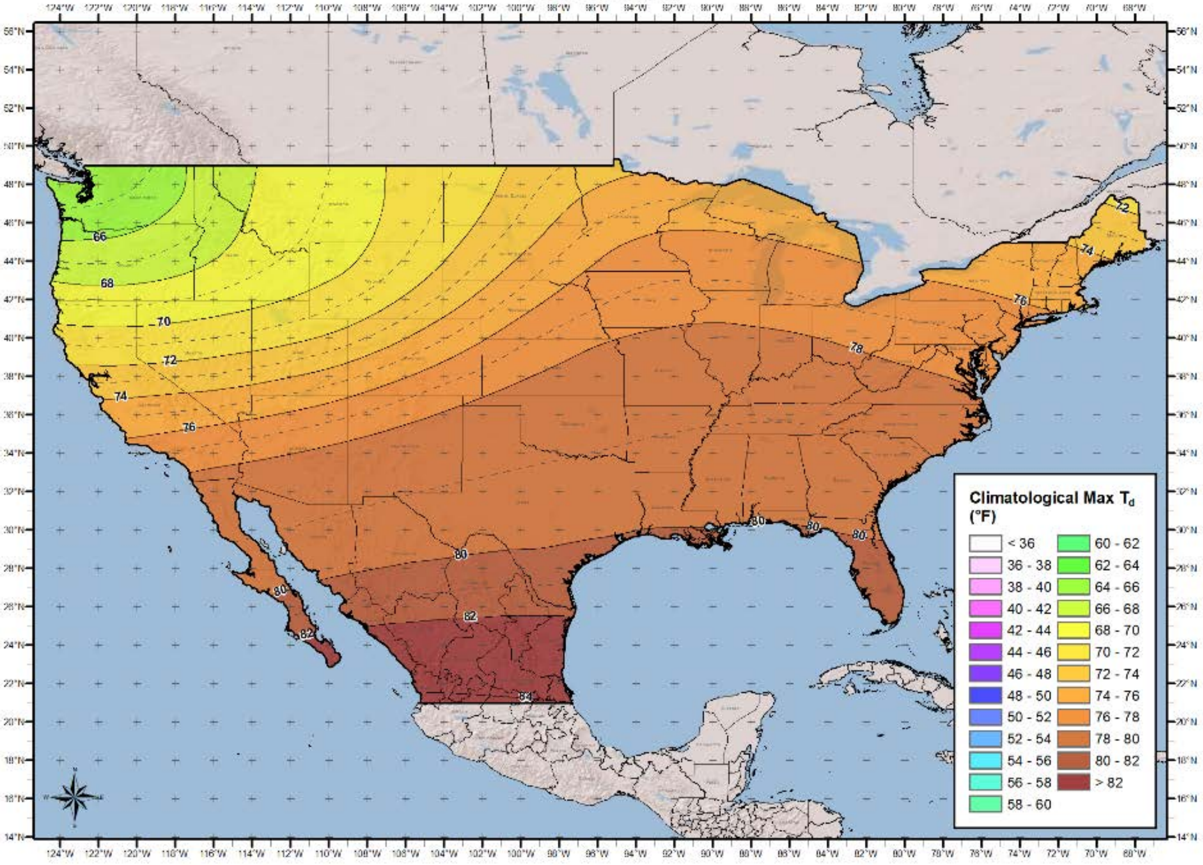
100-year Return Frequency 6-hour Maximum Dew Point Climatology
July (°F)



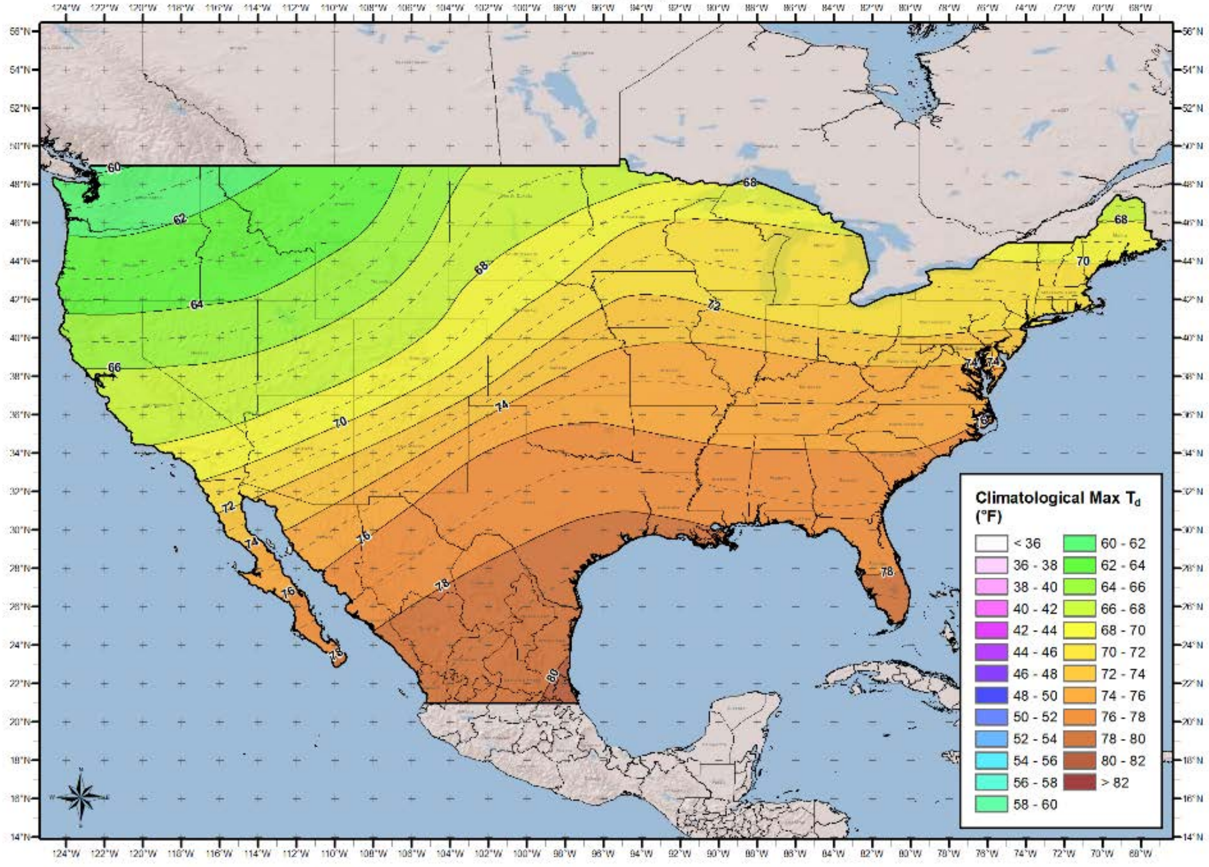
## 100-year Return Frequency 6-hour Maximum Dew Point Climatology August (°F)



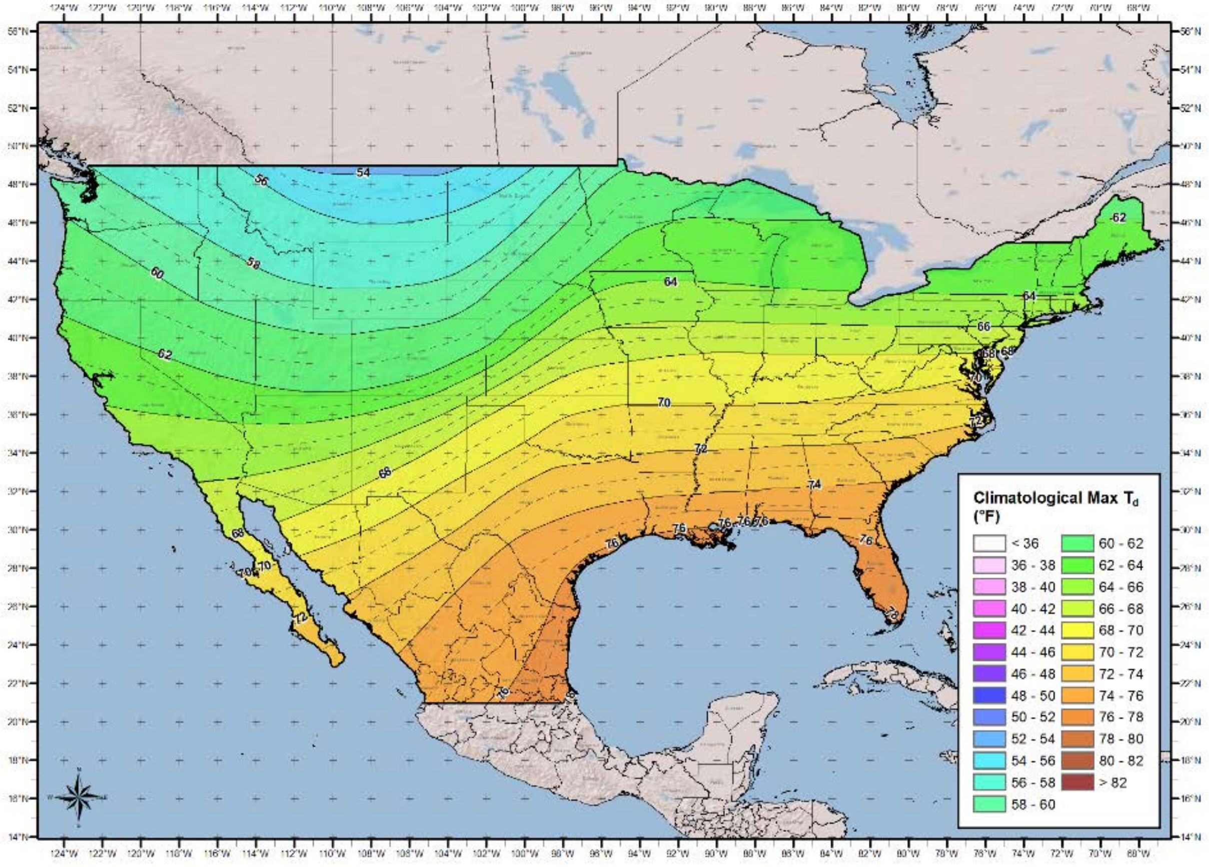
100-year Return Frequency 6-hour Maximum Dew Point Climatology
September (°F)



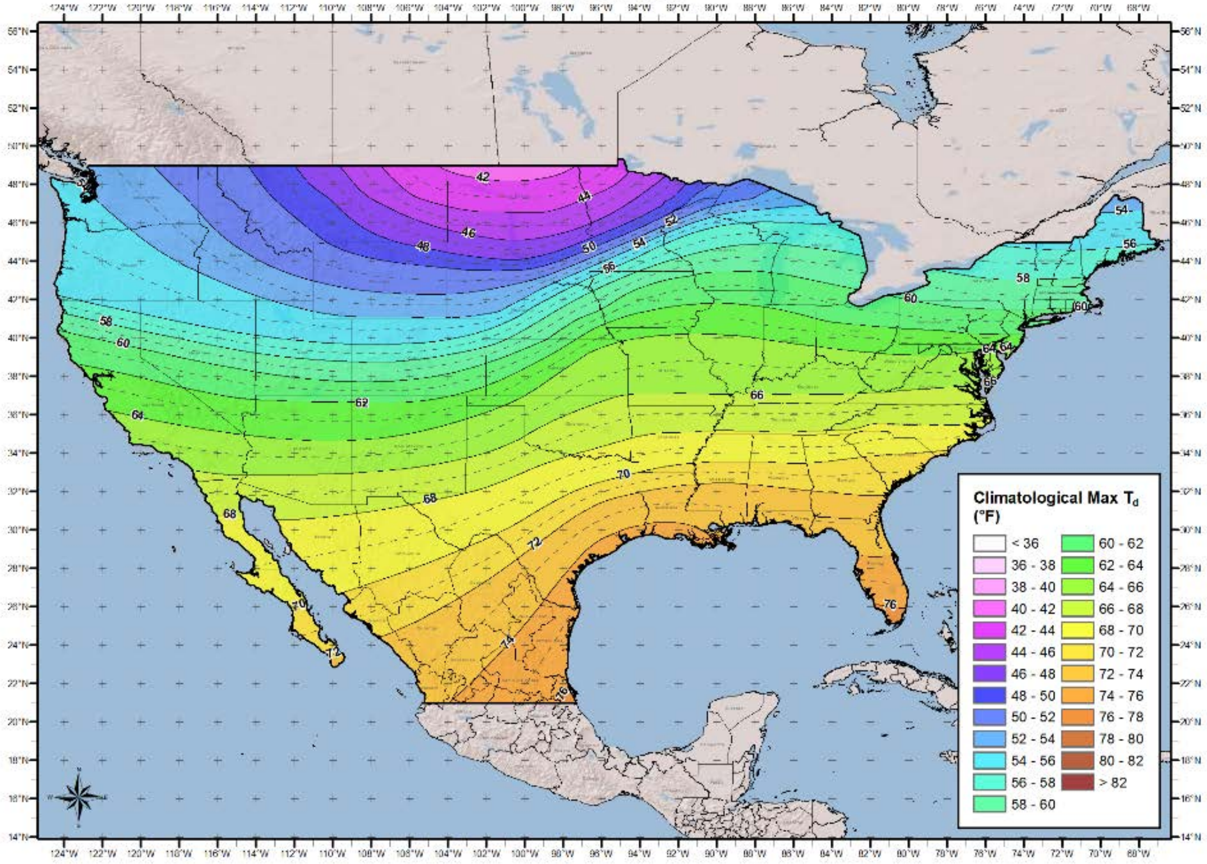
## 100-year Return Frequency 6-hour Maximum Dew Point Climatology
## October (°F)



## 100-year Return Frequency 6-hour Maximum Dew Point Climatology
## November (°F)

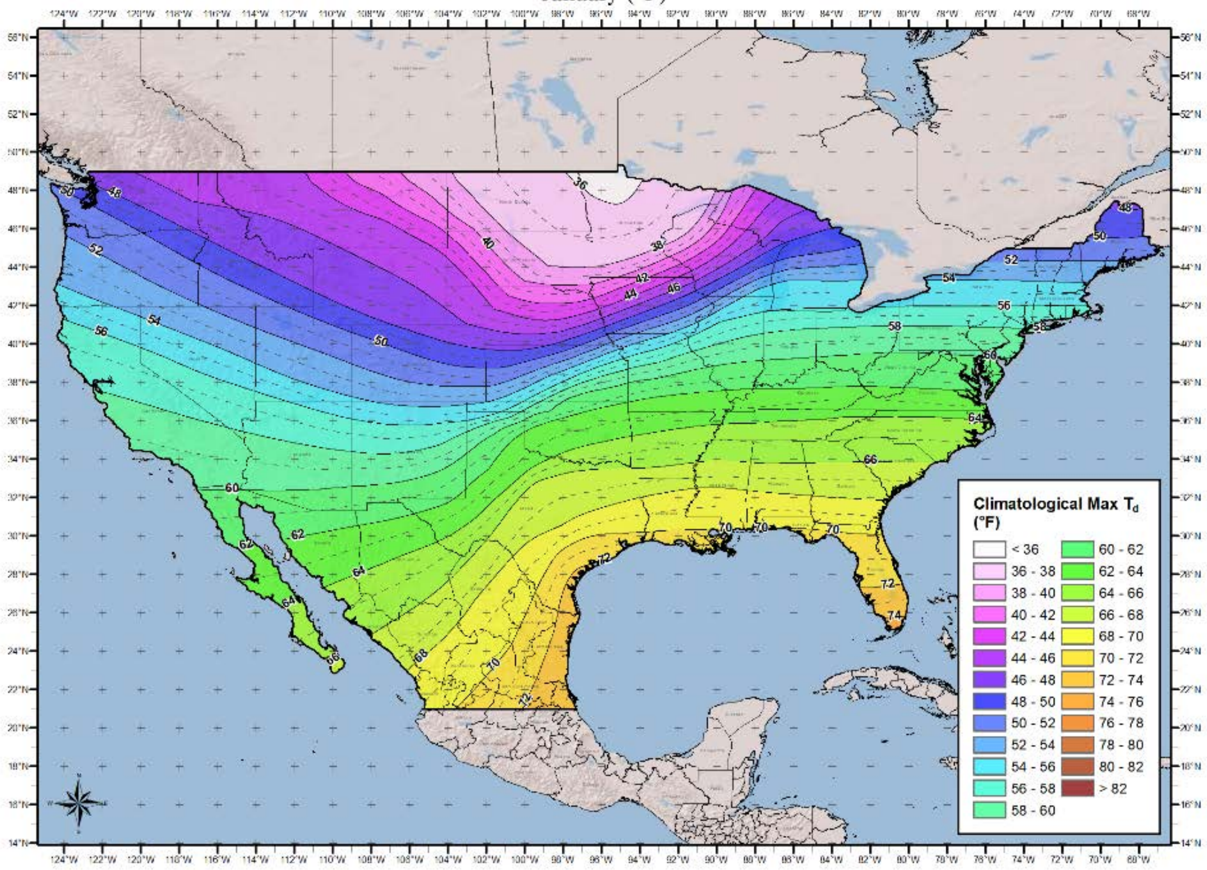


100-year Return Frequency 6-hour Maximum Dew Point Climatology
December (°F)

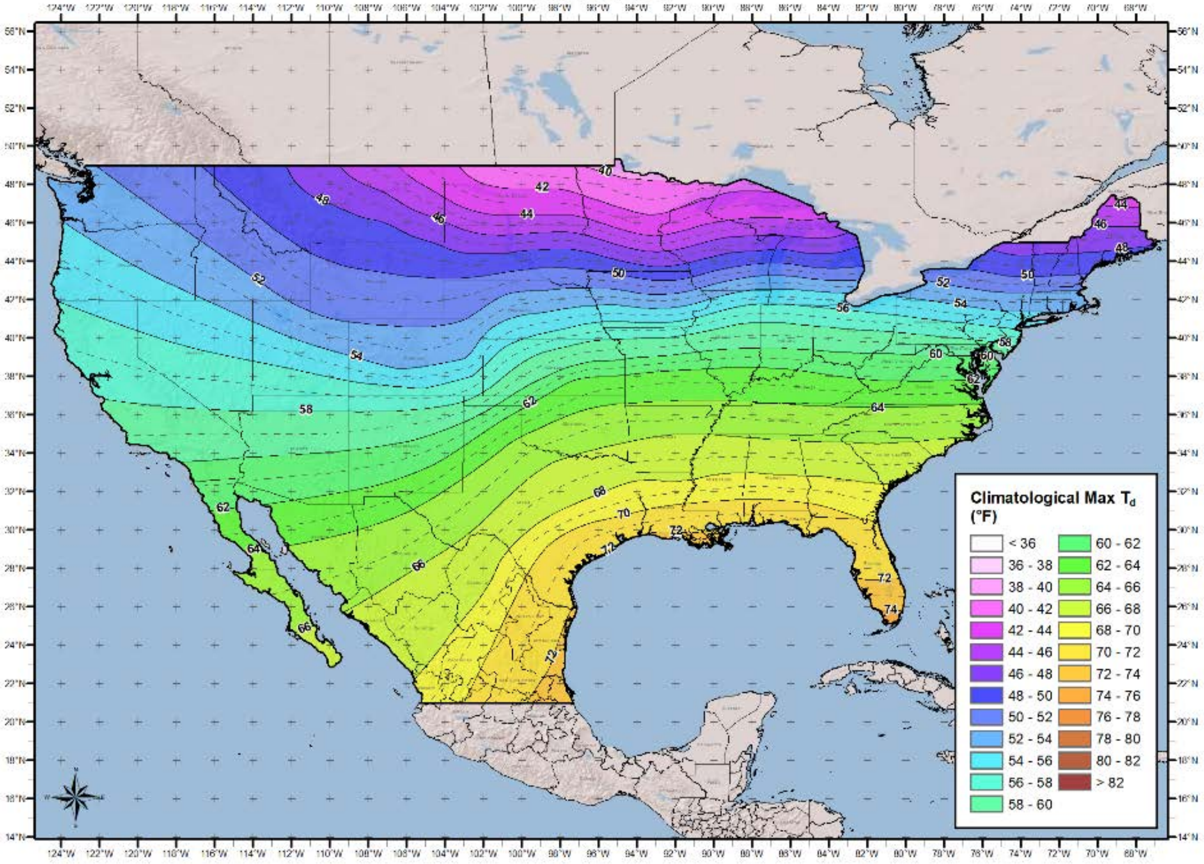


# 12-hour 1000mb Dew Point Maps

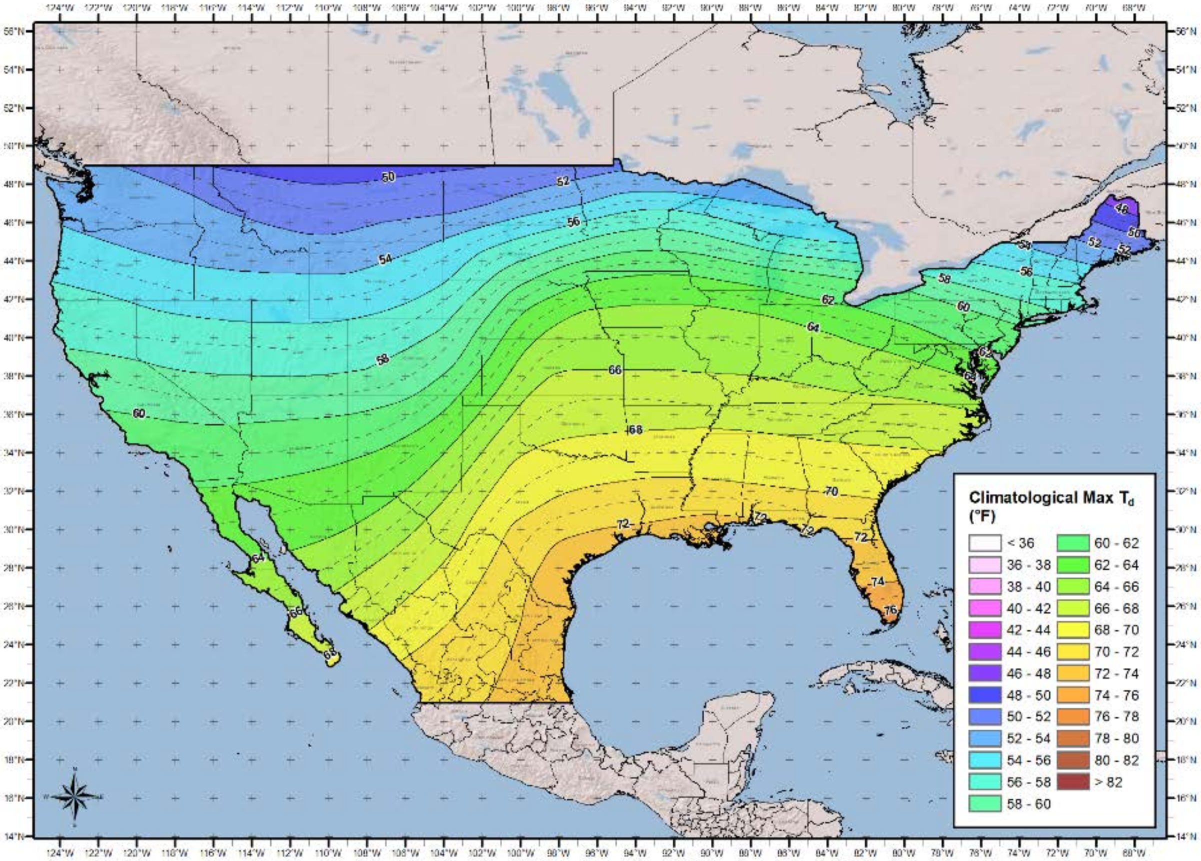
## 100-year Return Frequency 12-hour Maximum Dew Point Climatology
## January (°F)



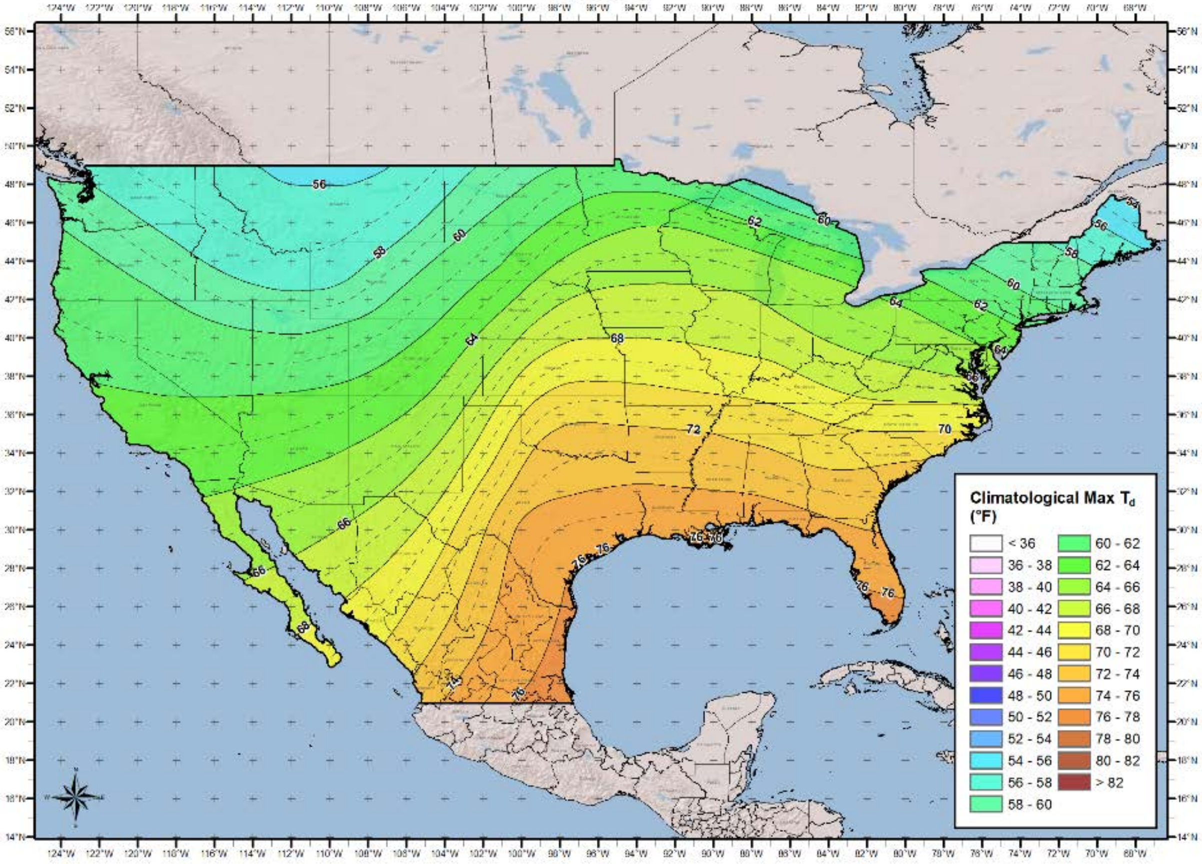
## 100-year Return Frequency 12-hour Maximum Dew Point Climatology
## February (°F)


124°W 122°W 120°W 118°W 116°W 114°W 112°W 110°W 108°W 106°W 104°W 102°W 100°W 98°W 96°W 94°W 92°W 90°W 88°W 86°W 84°W 82°W 80°W 78°W 76°W 74°W 72°W 70°W 68°W
56°N 54°N 52°N 50°N 48°N 46°N 44°N 42°N 40°N 38°N 36°N 34°N 32°N 30°N 28°N 26°N 24°N 22°N 20°N 18°N 16°N 14°N
40
42
44
46
48
50
52
54
56
58
60
62
64
66
68
70
72
74
Climatological Max $T_d$ (°F)
< 36
36 - 38
38 - 40
40 - 42
42 - 44
44 - 46
46 - 48
48 - 50
50 - 52
52 - 54
54 - 56
56 - 58
58 - 60
60 - 62
62 - 64
64 - 66
66 - 68
68 - 70
70 - 72
72 - 74
74 - 76
76 - 78
78 - 80
80 - 82
> 82

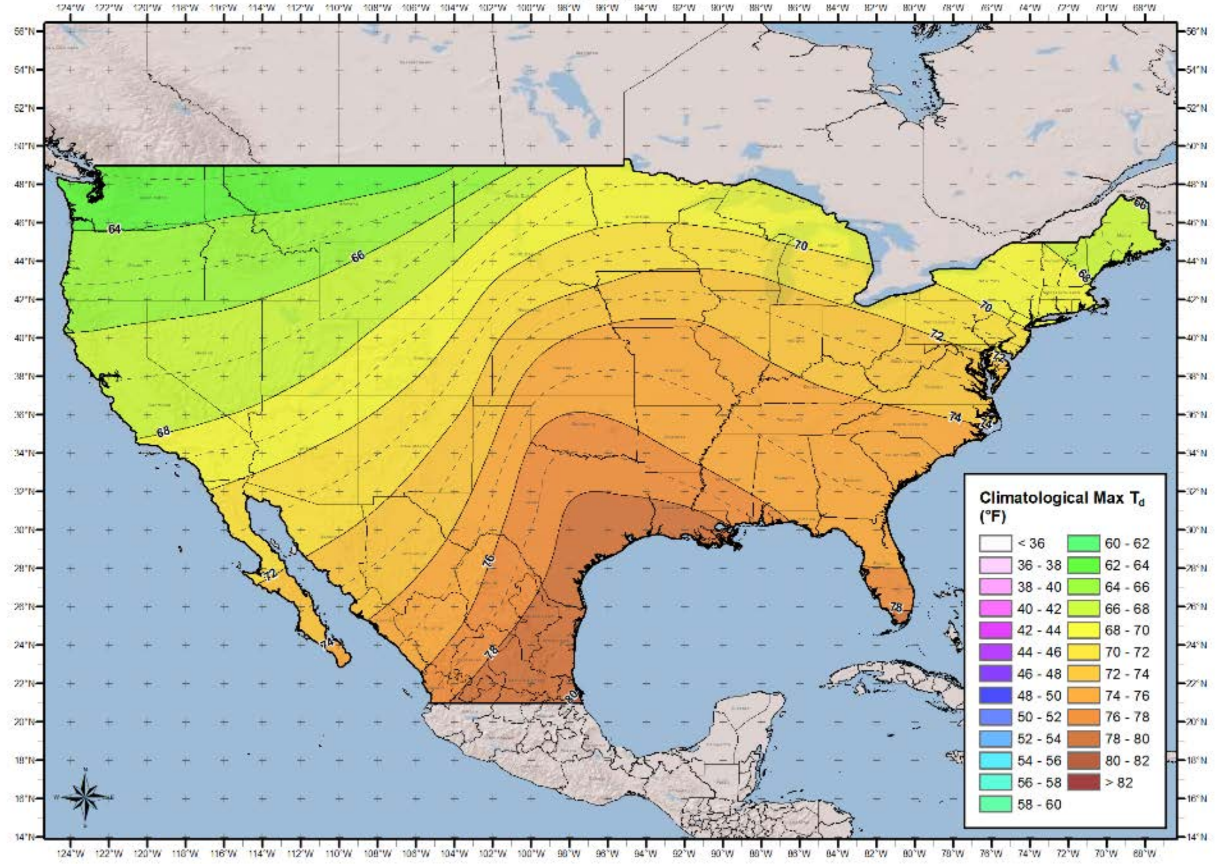
## 100-year Return Frequency 12-hour Maximum Dew Point Climatology
## March (°F)



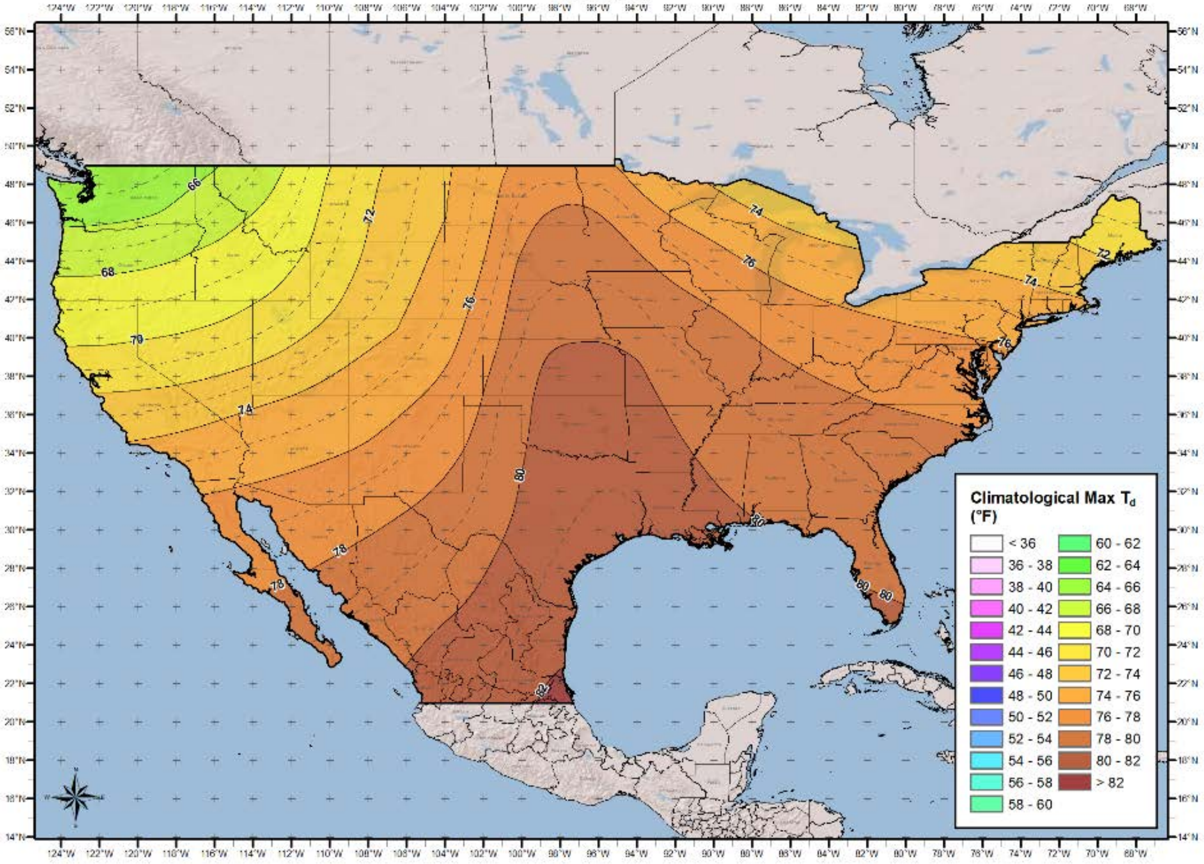
## 100-year Return Frequency 12-hour Maximum Dew Point Climatology
## April (°F)



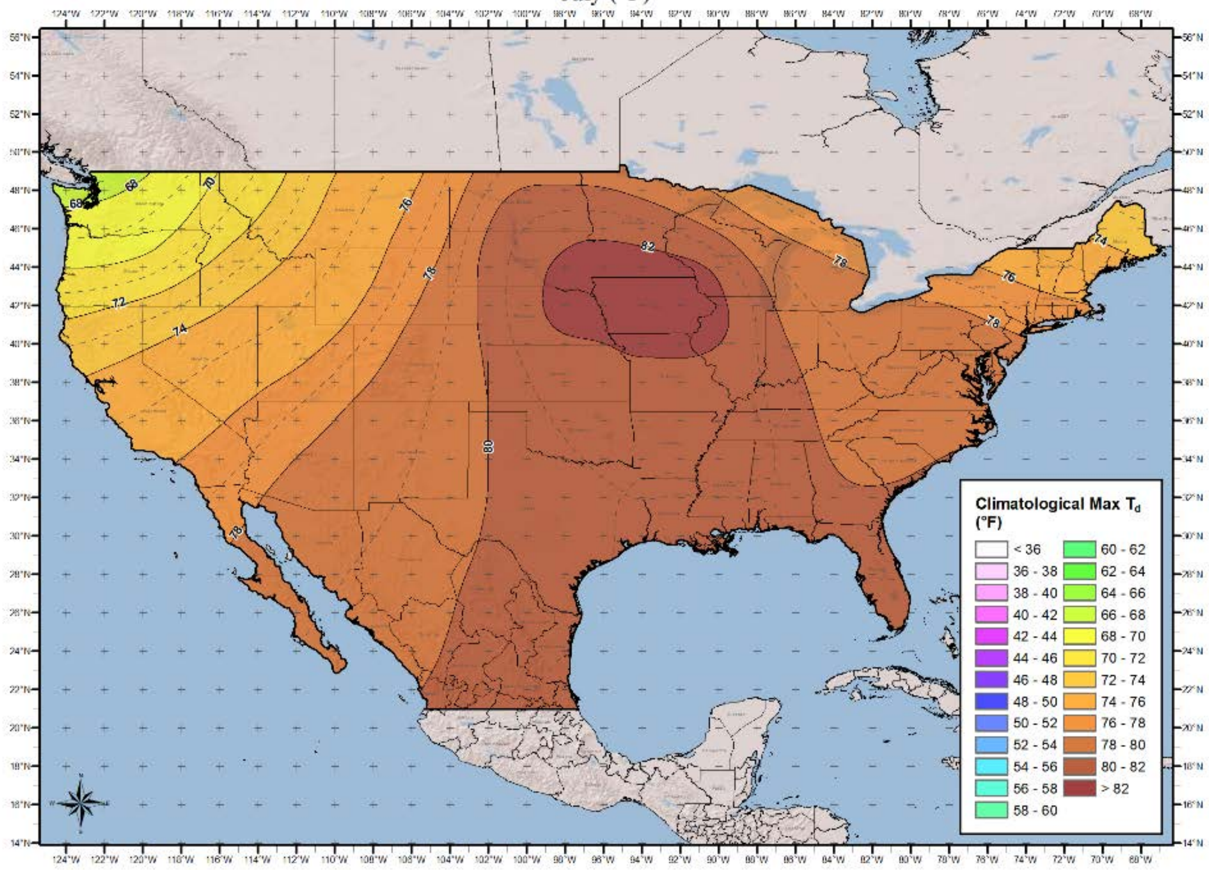
100-year Return Frequency 12-hour Maximum Dew Point Climatology
May (°F)



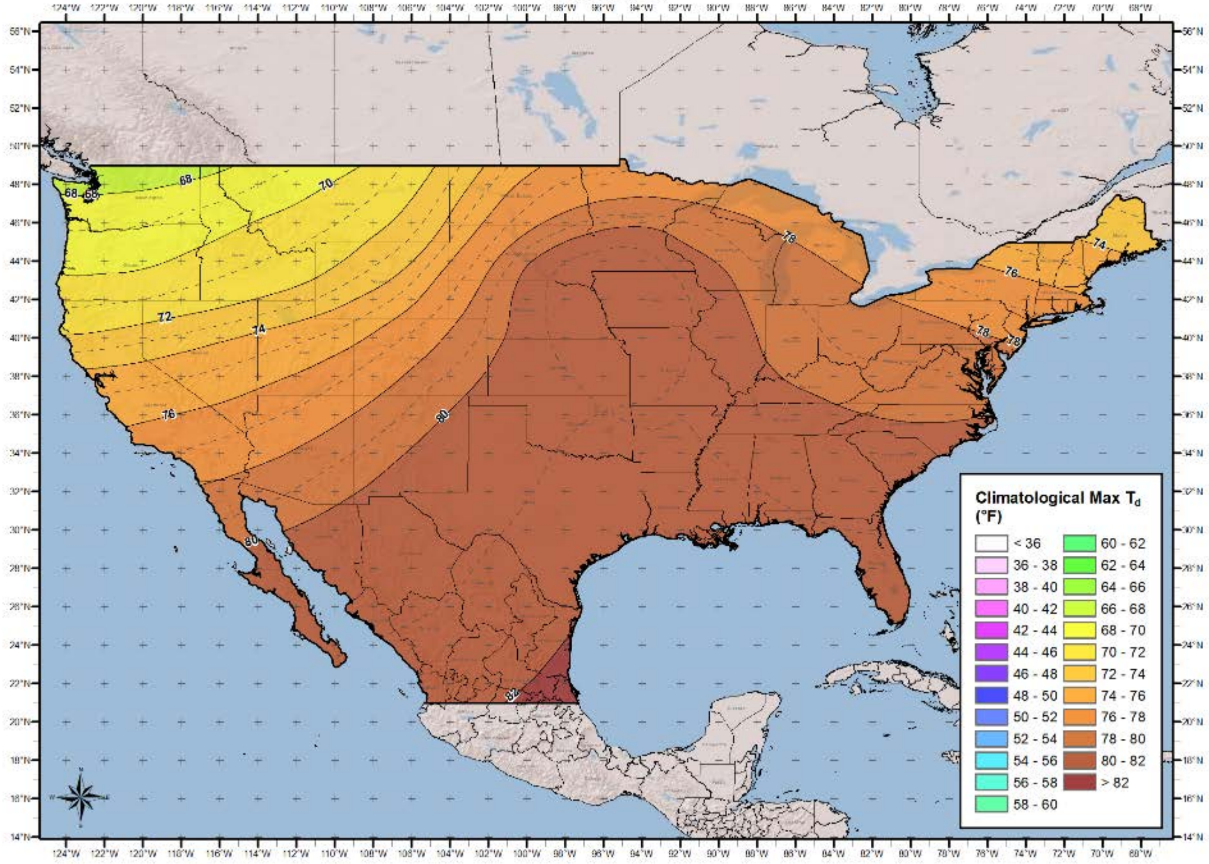
## 100-year Return Frequency 12-hour Maximum Dew Point Climatology
## June (°F)



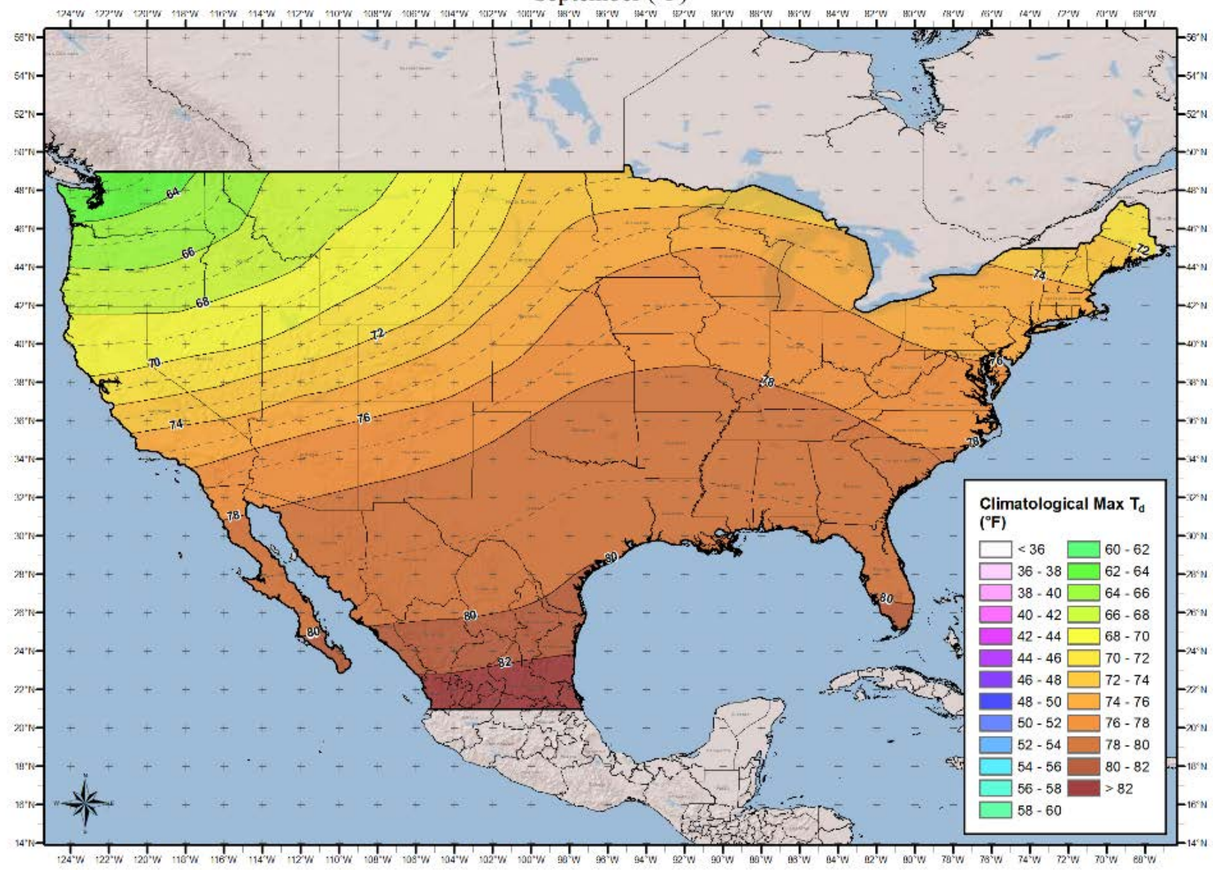
100-year Return Frequency 12-hour Maximum Dew Point Climatology
July (°F)



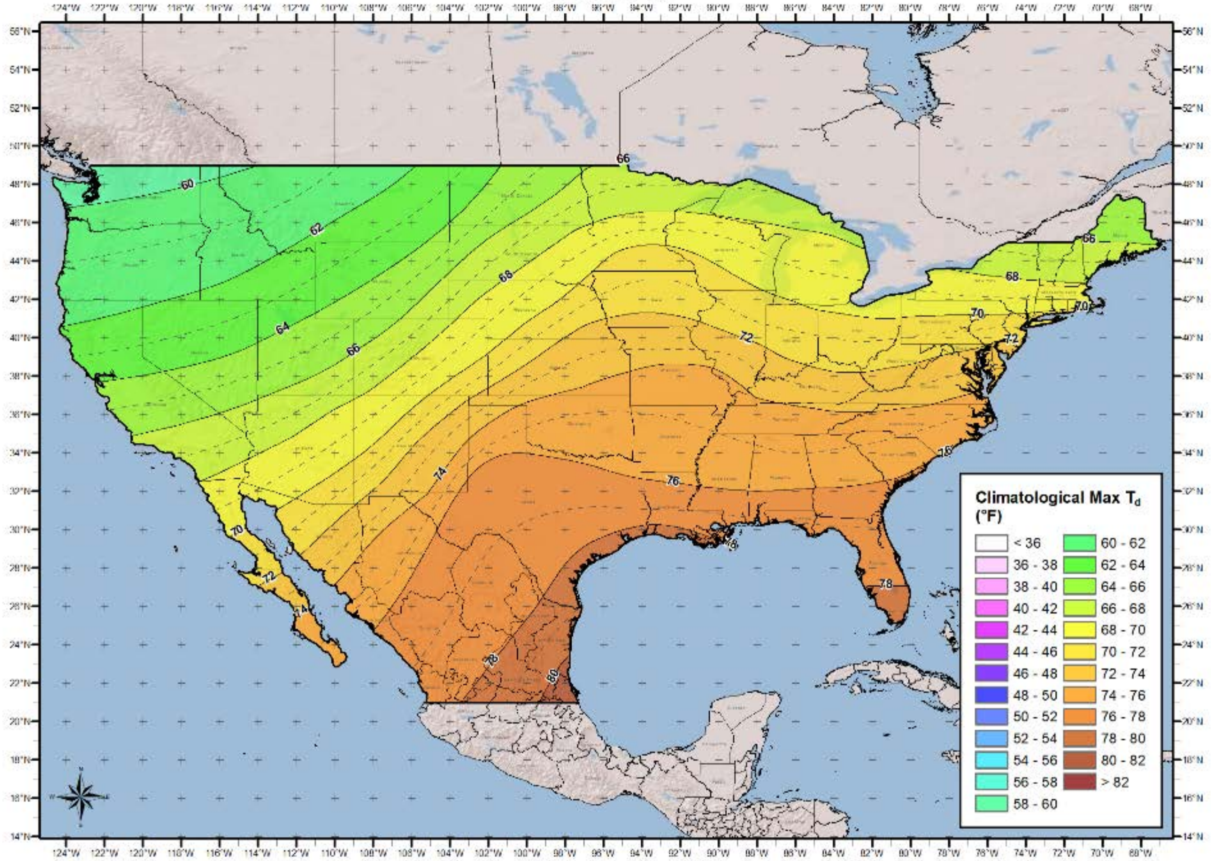
100-year Return Frequency 12-hour Maximum Dew Point Climatology
August (°F)



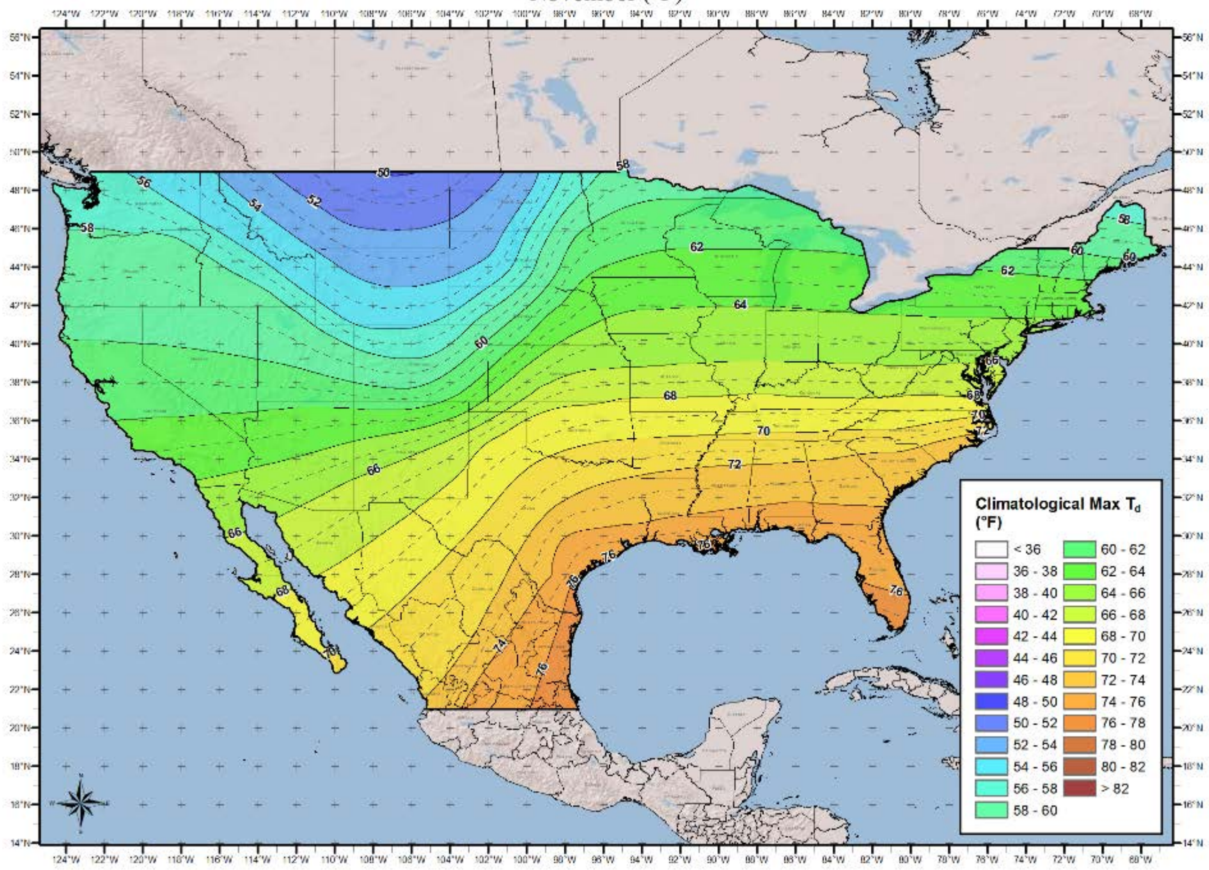
## 100-year Return Frequency 12-hour Maximum Dew Point Climatology September (°F)



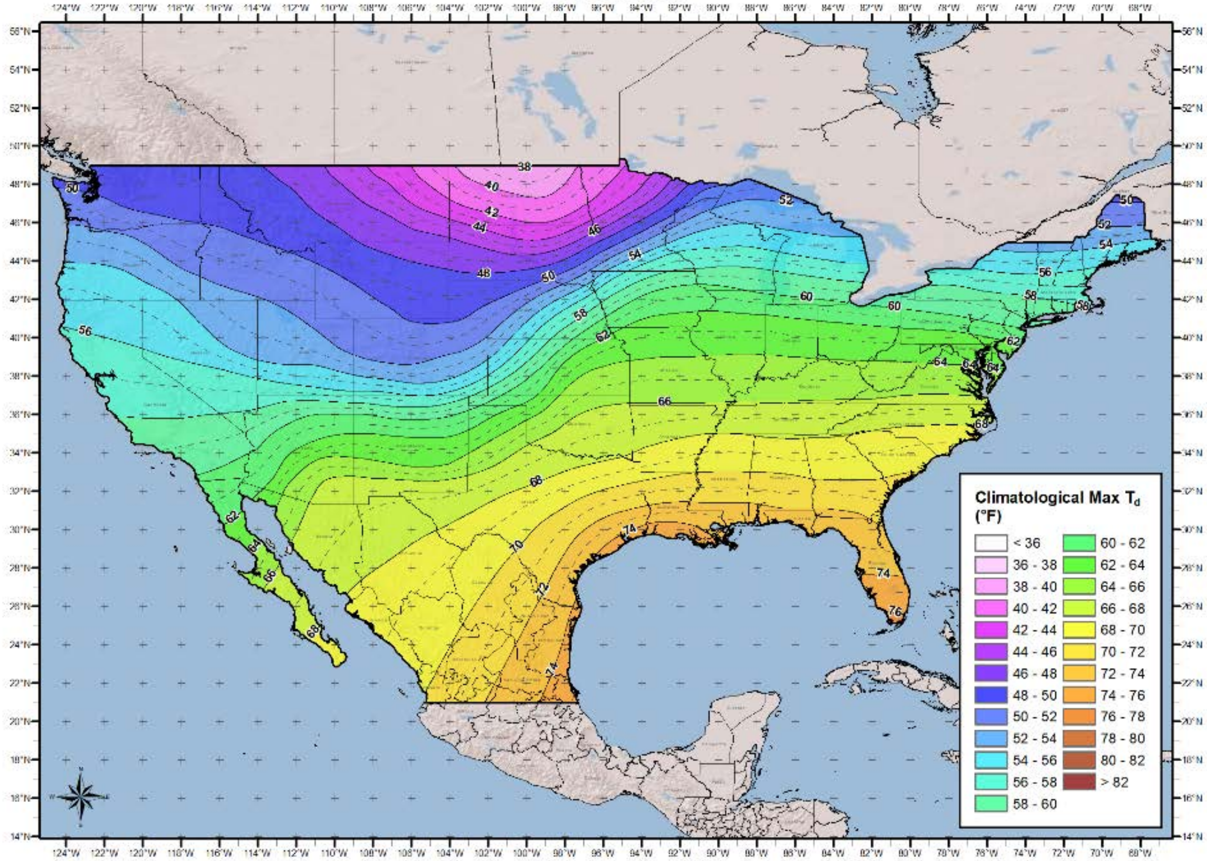
## 100-year Return Frequency 12-hour Maximum Dew Point Climatology
October (°F)



# 100-year Return Frequency 12-hour Maximum Dew Point Climatology
## November (°F)

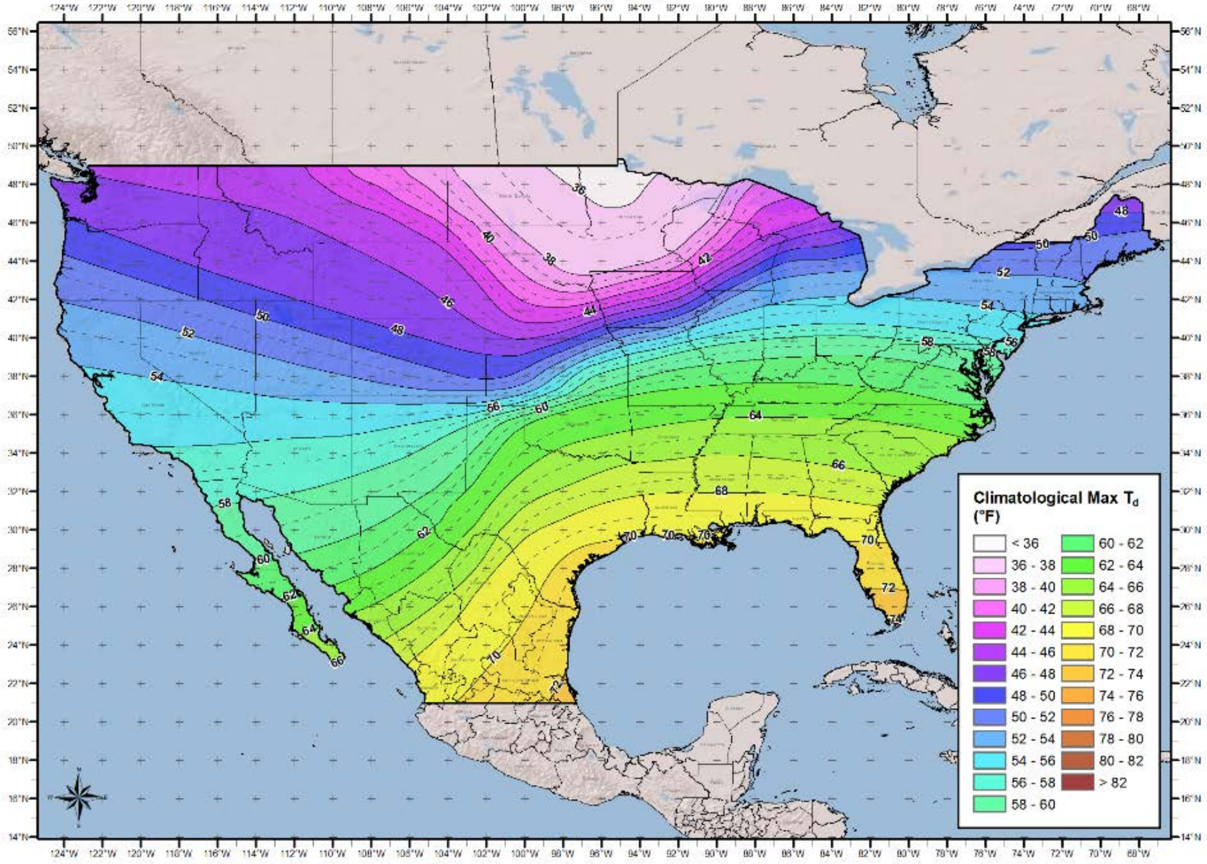


## 100-year Return Frequency 12-hour Maximum Dew Point Climatology December (°F)

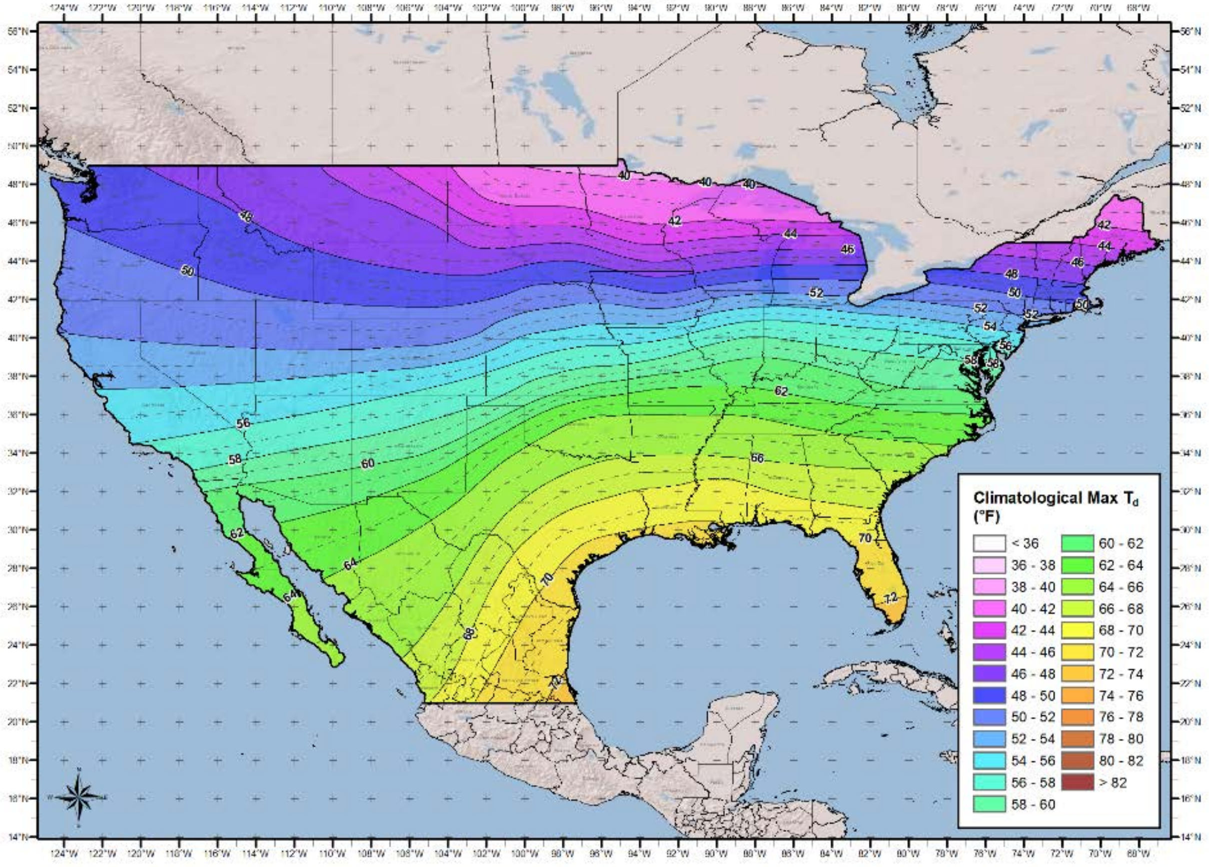


# 24-hour 1000mb Dew Point Maps

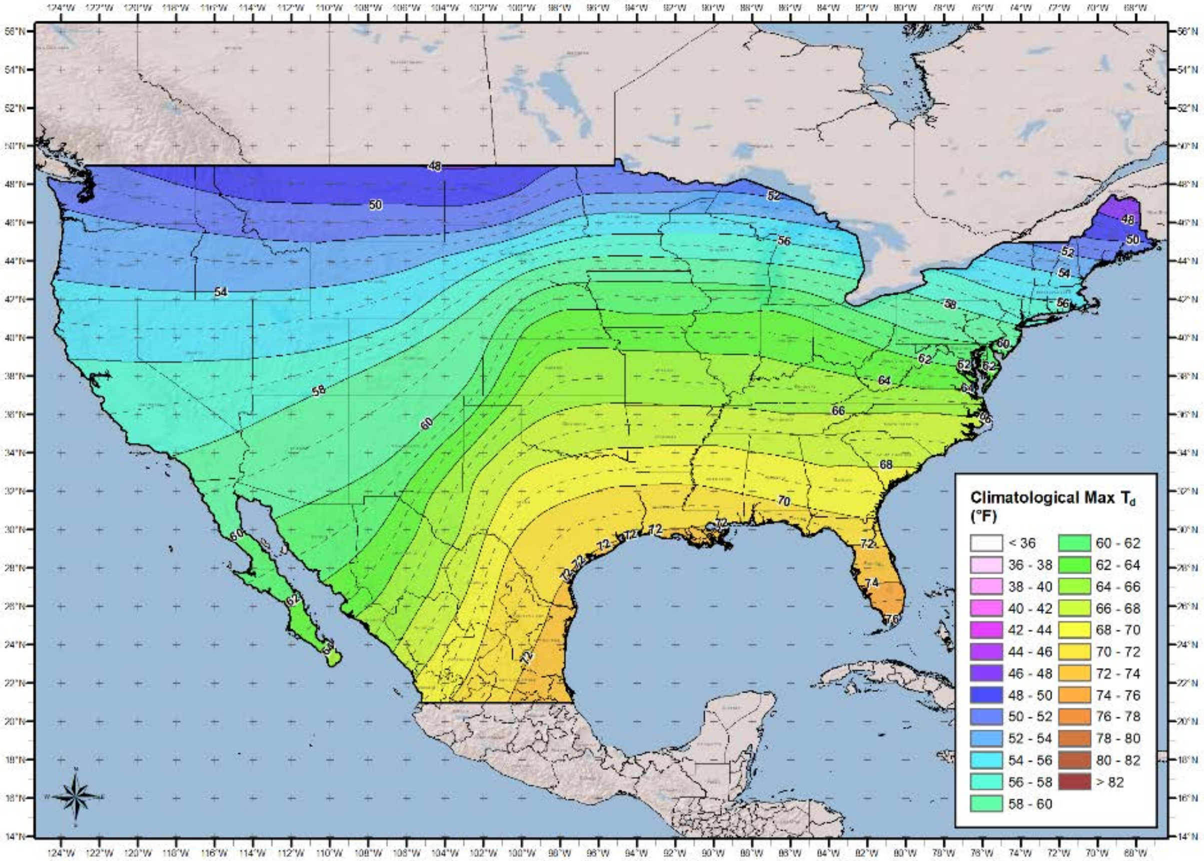
## 100-year Return Frequency 24-hour Maximum Dew Point Climatology
January (°F)



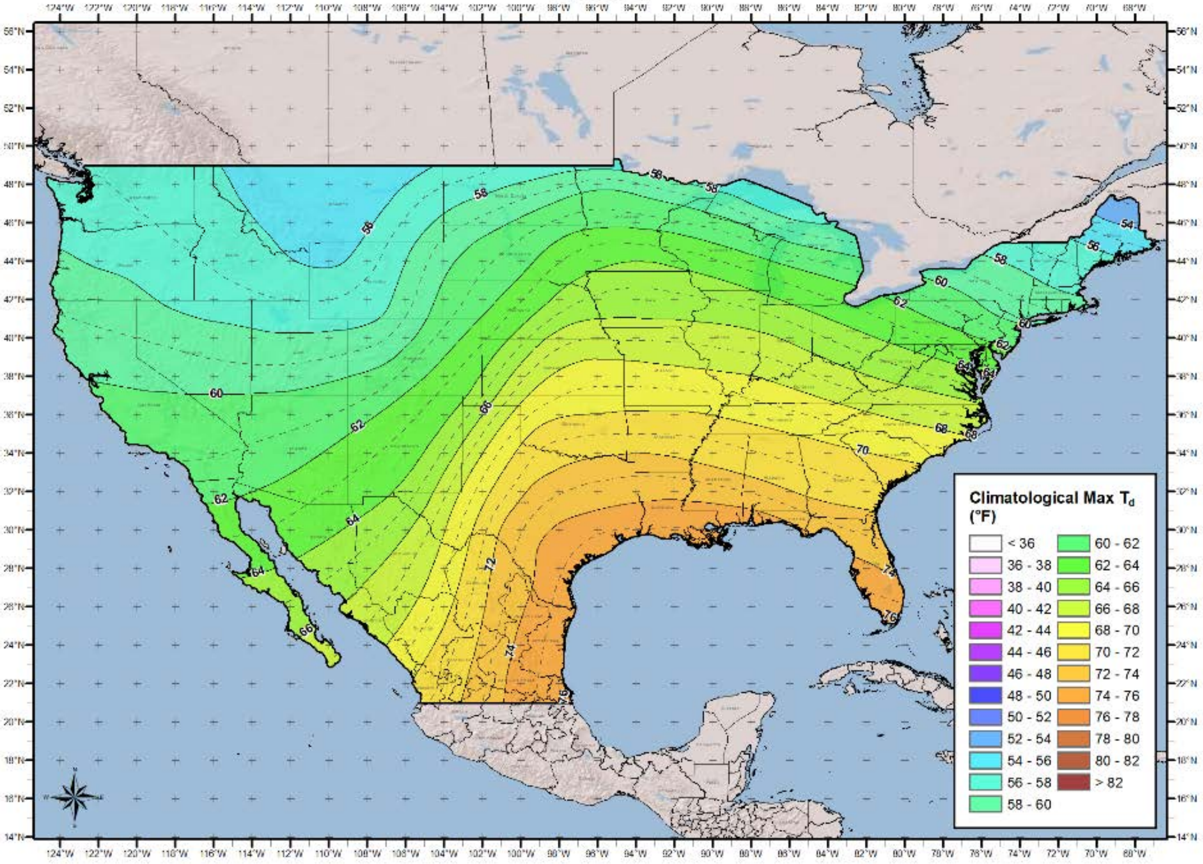
## 100-year Return Frequency 24-hour Maximum Dew Point Climatology February (°F)



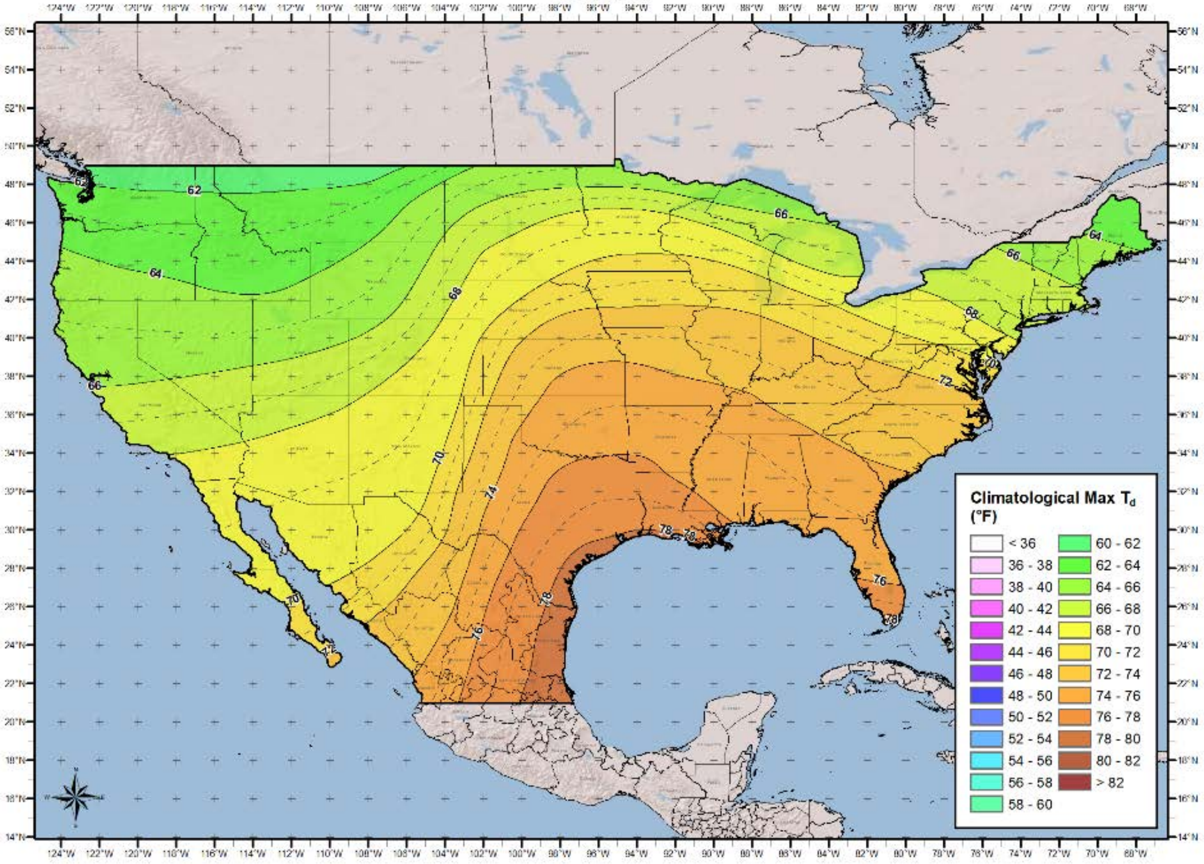
## 100-year Return Frequency 24-hour Maximum Dew Point Climatology
## March (°F)


124°W 122°W 120°W 118°W 116°W 114°W 112°W 110°W 108°W 106°W 104°W 102°W 100°W 98°W 96°W 94°W 92°W 90°W 88°W 86°W 84°W 82°W 80°W 78°W 76°W 74°W 72°W 70°W 68°W
56°N 54°N 52°N 50°N 48°N 46°N 44°N 42°N 40°N 38°N 36°N 34°N 32°N 30°N 28°N 26°N 24°N 22°N 20°N 18°N 16°N 14°N
48
50
52
54
56
58
60
62
64
66
68
70
72
74
76
Climatological Max $T_d$ (°F)
< 36
36 - 38
38 - 40
40 - 42
42 - 44
44 - 46
46 - 48
48 - 50
50 - 52
52 - 54
54 - 56
56 - 58
58 - 60
60 - 62
62 - 64
64 - 66
66 - 68
68 - 70
70 - 72
72 - 74
74 - 76
76 - 78
78 - 80
80 - 82
> 82

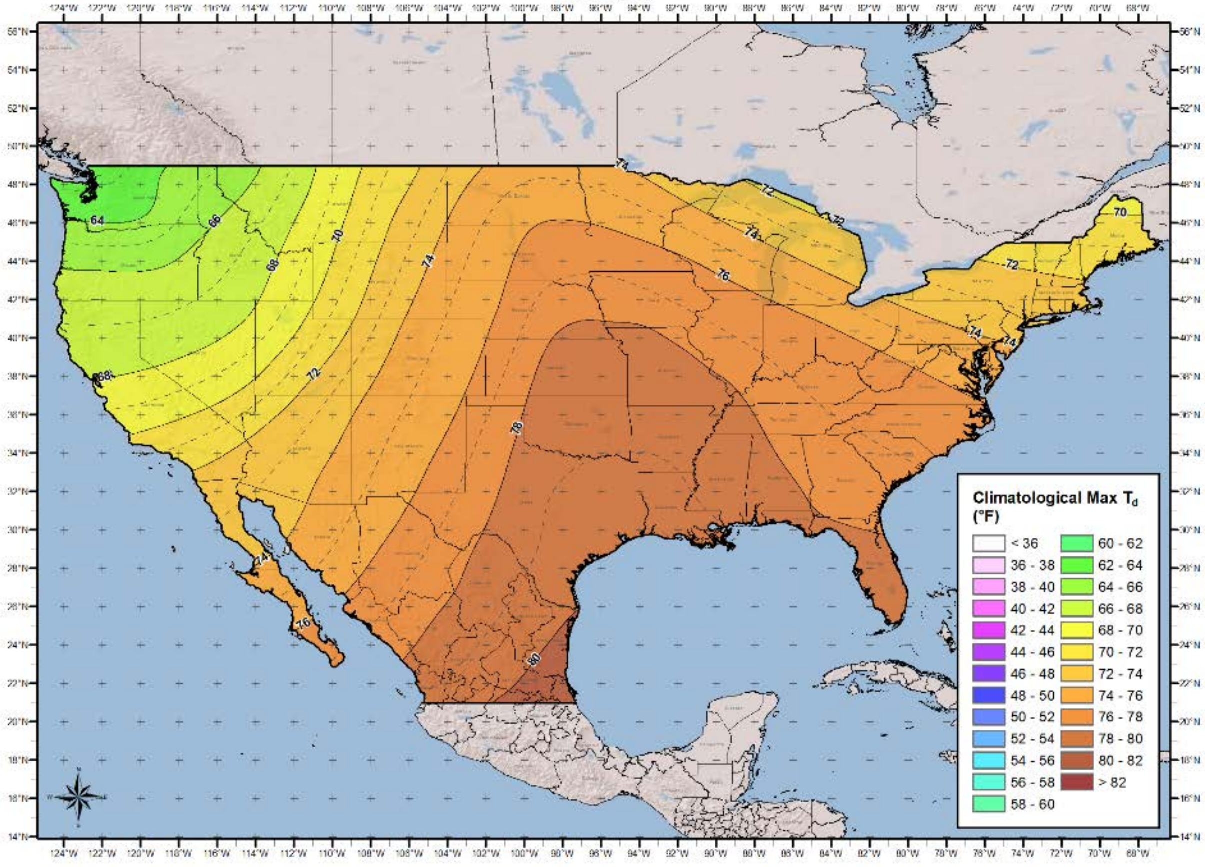
100-year Return Frequency 24-hour Maximum Dew Point Climatology
April (°F)


124°W 122°W 120°W 118°W 116°W 114°W 112°W 110°W 108°W 106°W 104°W 102°W 100°W 98°W 96°W 94°W 92°W 90°W 88°W 86°W 84°W 82°W 80°W 78°W 76°W 74°W 72°W 70°W 68°W
56°N 54°N 52°N 50°N 48°N 46°N 44°N 42°N 40°N 38°N 36°N 34°N 32°N 30°N 28°N 26°N 24°N 22°N 20°N 18°N 16°N 14°N
58
58
56
58
54
56
58
60
62
60
62
64
66
68
68
70
74
76
60
62
64
66
62
64
66
72
74
76
Climatological Max $T_d$ (°F)
⬜ < 36
🟪 36 - 38
🟪 38 - 40
🟪 40 - 42
🟪 42 - 44
🟪 44 - 46
🟪 46 - 48
🟦 48 - 50
🟦 50 - 52
🟦 52 - 54
🟦 54 - 56
🟩 56 - 58
🟩 58 - 60
🟩 60 - 62
🟩 62 - 64
🟩 64 - 66
🟨 66 - 68
🟨 68 - 70
🟨 70 - 72
🟧 72 - 74
🟧 74 - 76
🟧 76 - 78
🟫 78 - 80
🟫 80 - 82
🟥 > 82

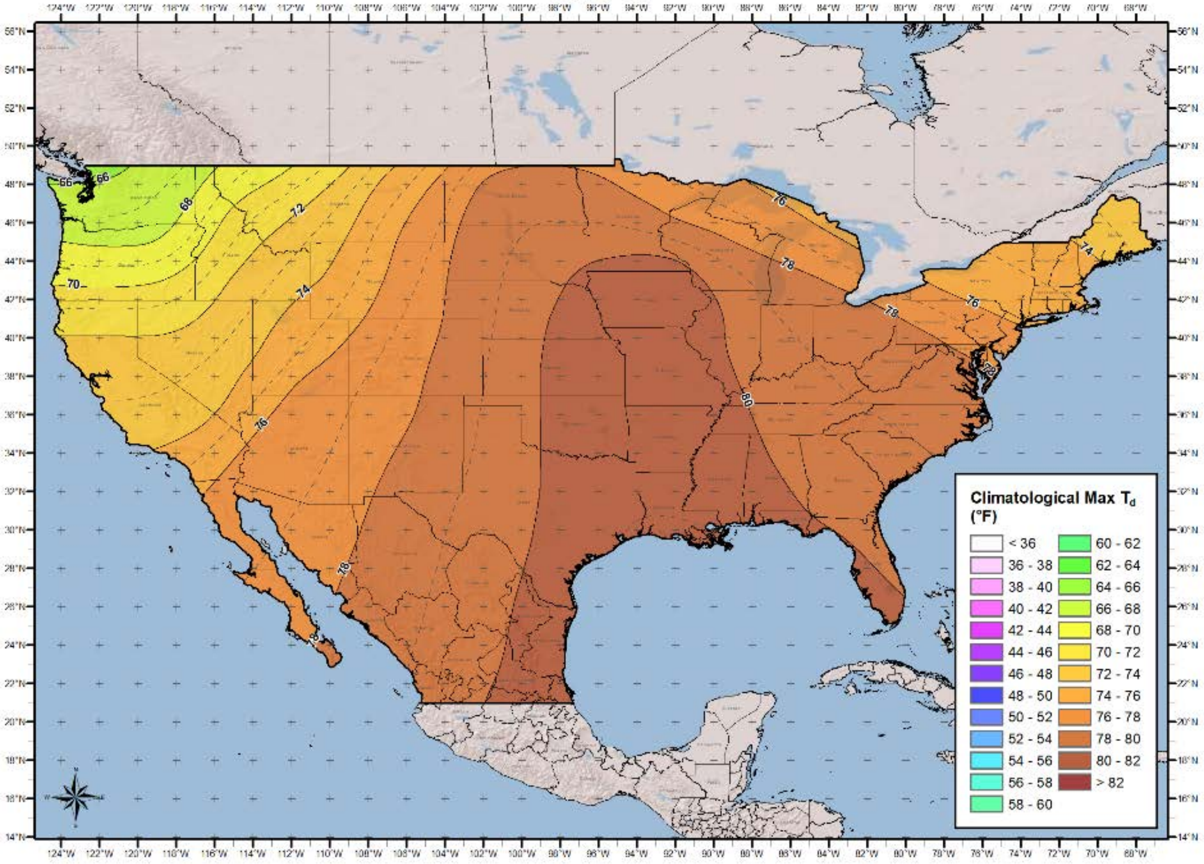
100-year Return Frequency 24-hour Maximum Dew Point Climatology
May (°F)



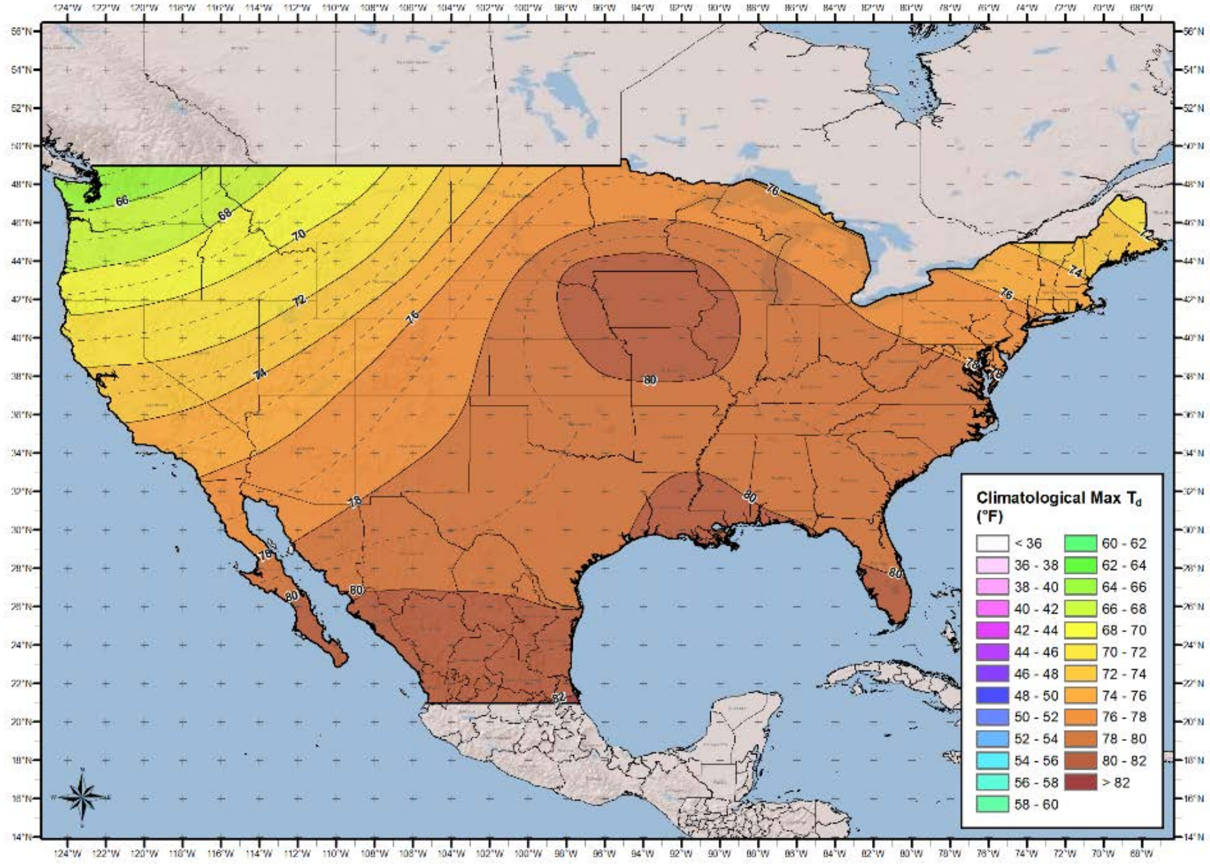
100-year Return Frequency 24-hour Maximum Dew Point Climatology
June (°F)



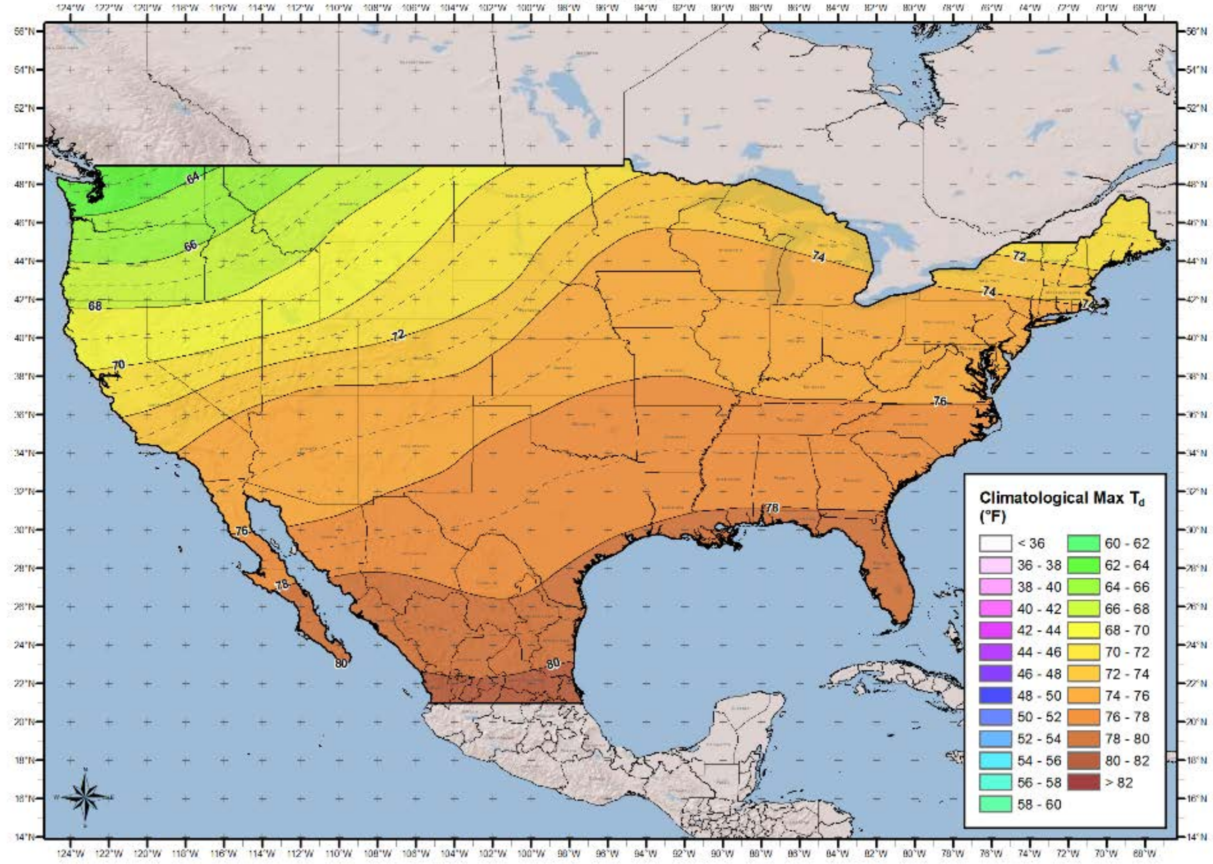
100-year Return Frequency 24-hour Maximum Dew Point Climatology
July (°F)



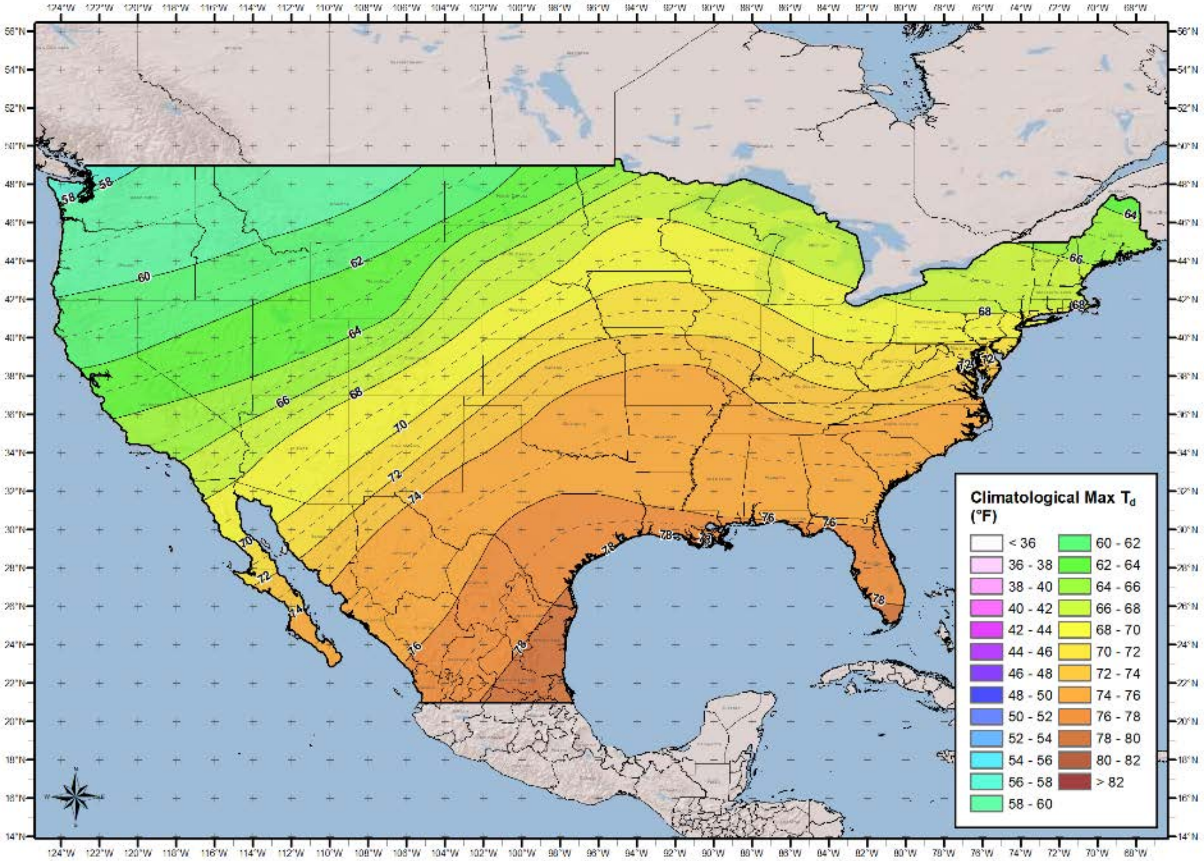
100-year Return Frequency 24-hour Maximum Dew Point Climatology
August (°F)



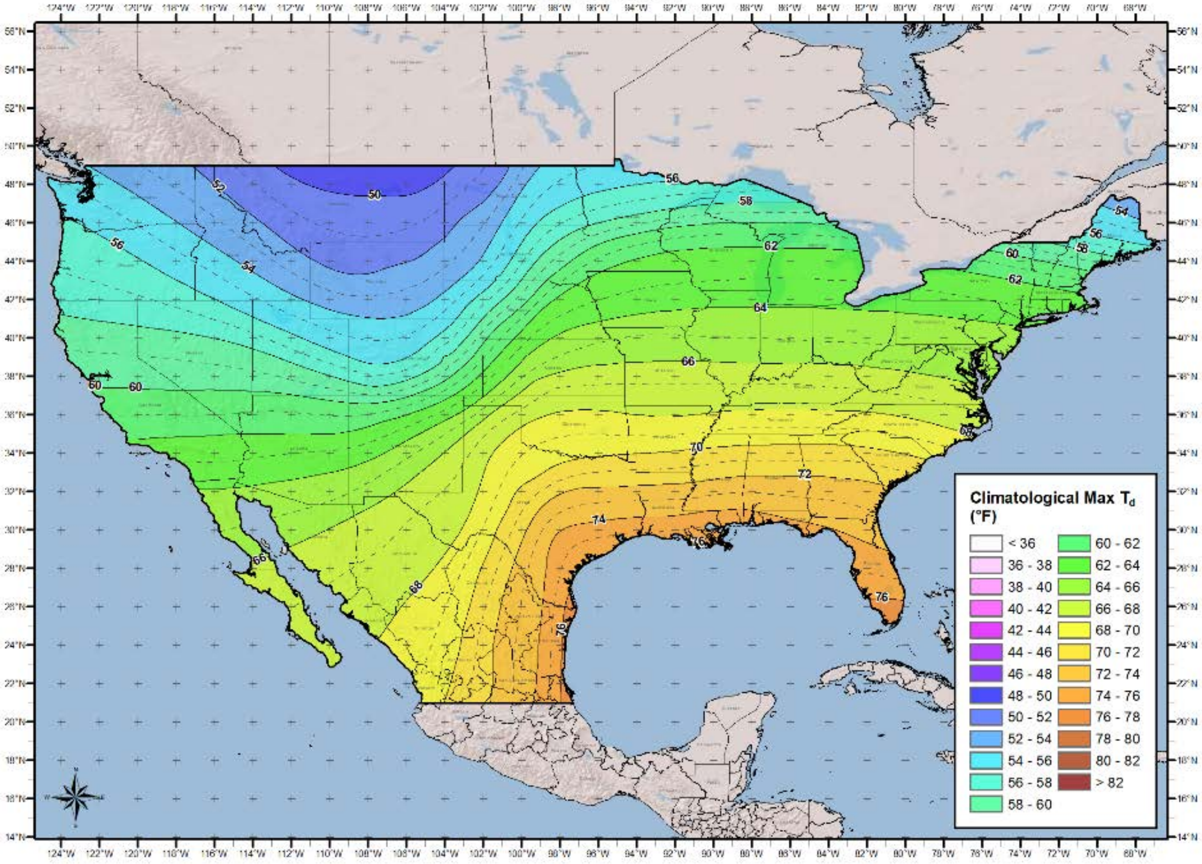
100-year Return Frequency 24-hour Maximum Dew Point Climatology
September (°F)



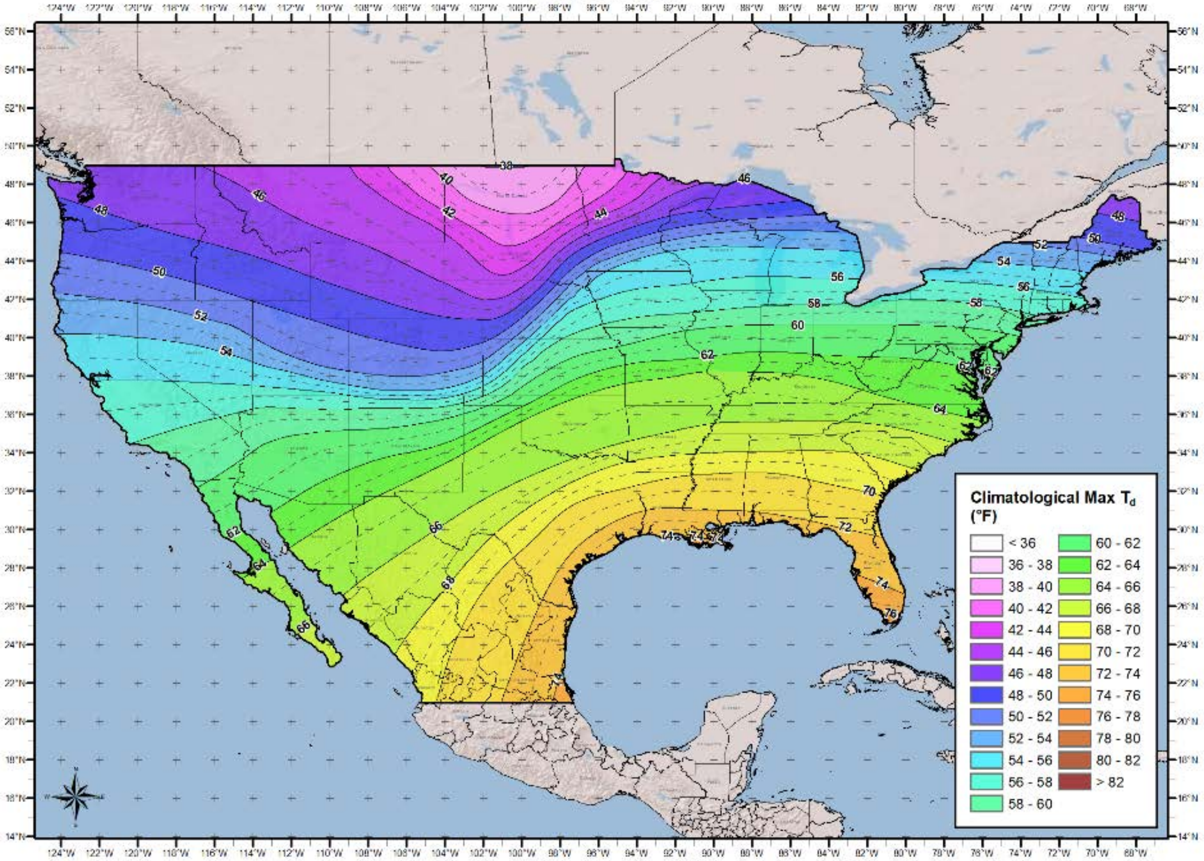
## 100-year Return Frequency 24-hour Maximum Dew Point Climatology
## October (°F)



100-year Return Frequency 24-hour Maximum Dew Point Climatology
November (°F)


Climatological Max $T_d$ (°F)
< 36
36 - 38
38 - 40
40 - 42
42 - 44
44 - 46
46 - 48
48 - 50
50 - 52
52 - 54
54 - 56
56 - 58
58 - 60
60 - 62
62 - 64
64 - 66
66 - 68
68 - 70
70 - 72
72 - 74
74 - 76
76 - 78
78 - 80
80 - 82
> 82
50
52
54
56
58
60
62
64
66
68
70
72
74
76
124°W 122°W 120°W 118°W 116°W 114°W 112°W 110°W 108°W 106°W 104°W 102°W 100°W 98°W 96°W 94°W 92°W 90°W 88°W 86°W 84°W 82°W 80°W 78°W 76°W 74°W 72°W 70°W 68°W
56°N 54°N 52°N 50°N 48°N 46°N 44°N 42°N 40°N 38°N 36°N 34°N 32°N 30°N 28°N 26°N 24°N 22°N 20°N 18°N 16°N 14°N

## 100-year Return Frequency 24-hour Maximum Dew Point Climatology
## December (°F)

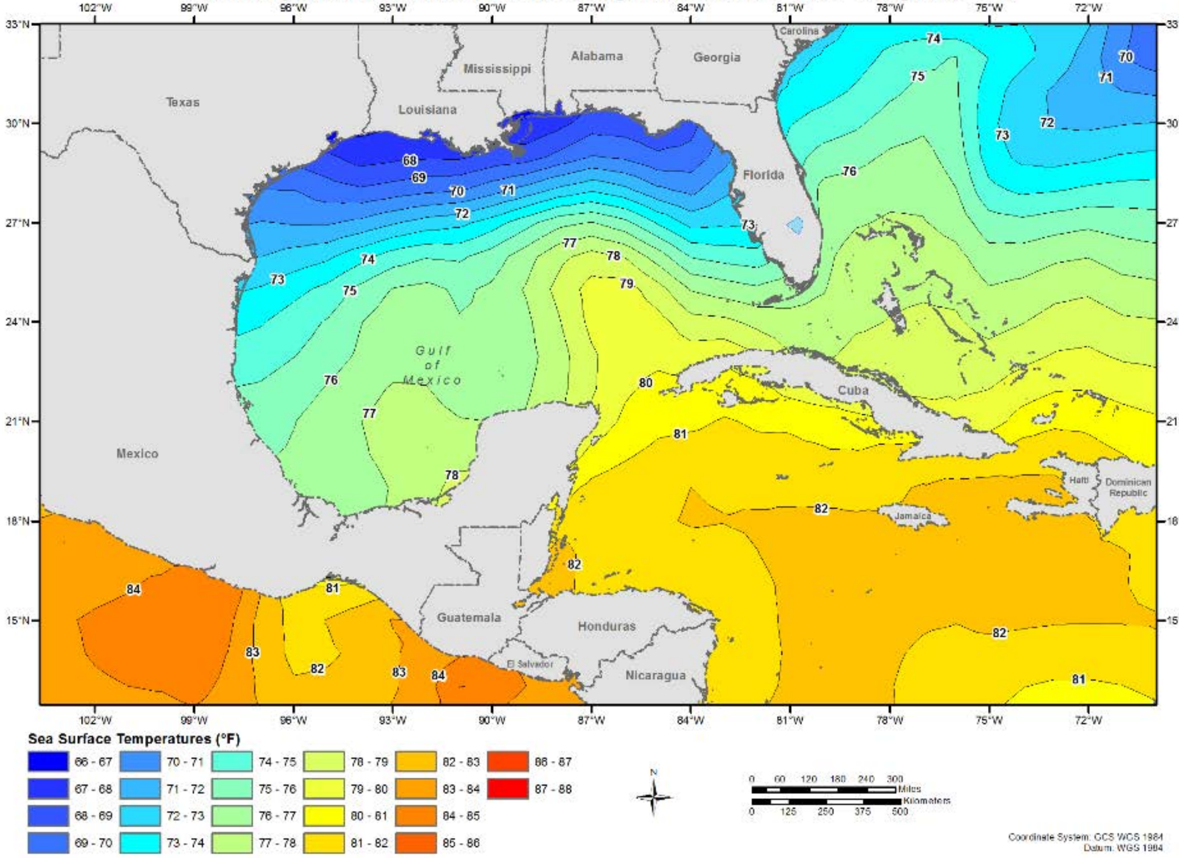


# Appendix D
## Sea Surface Temperature (SST) Climatology Maps

# 2-Sigma Sea Surface Temperature Maps

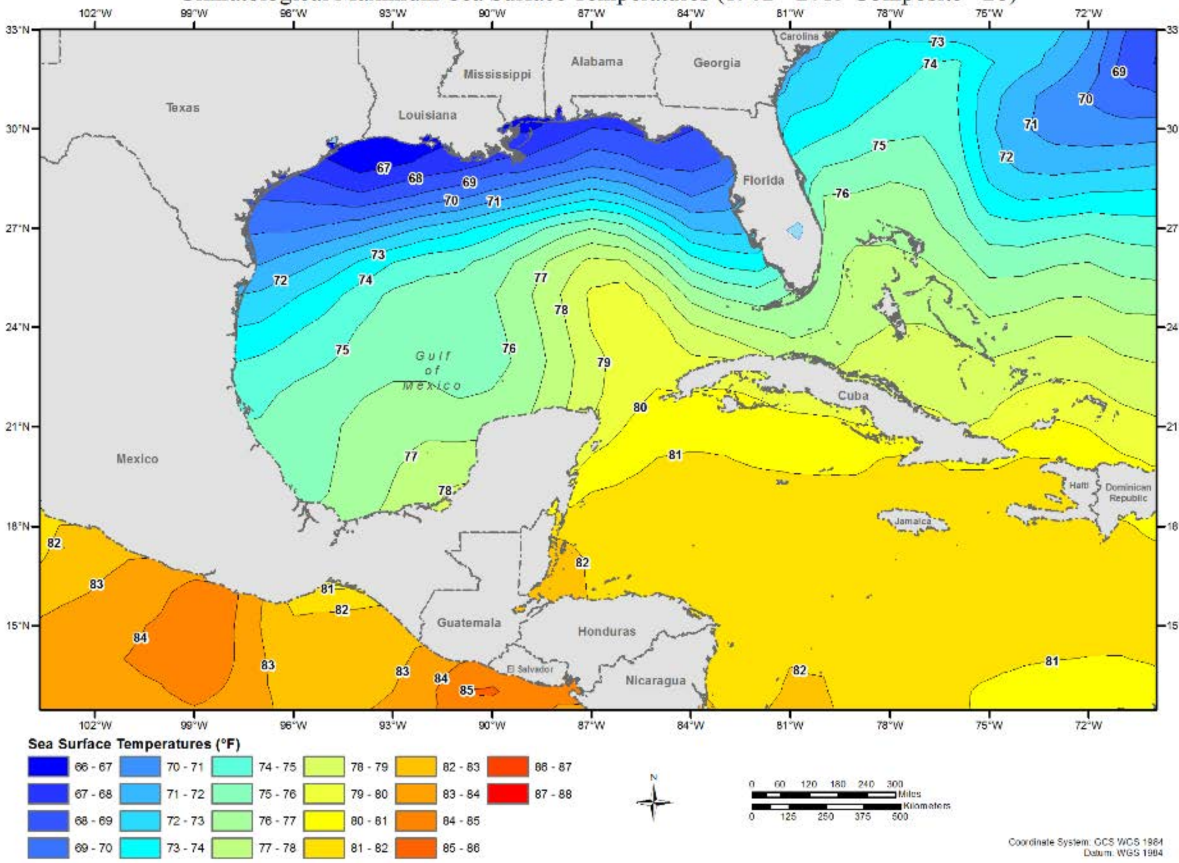
## January

Climatological Maximum Sea Surface Temperatures (1982 - 2013 Composite +2σ)



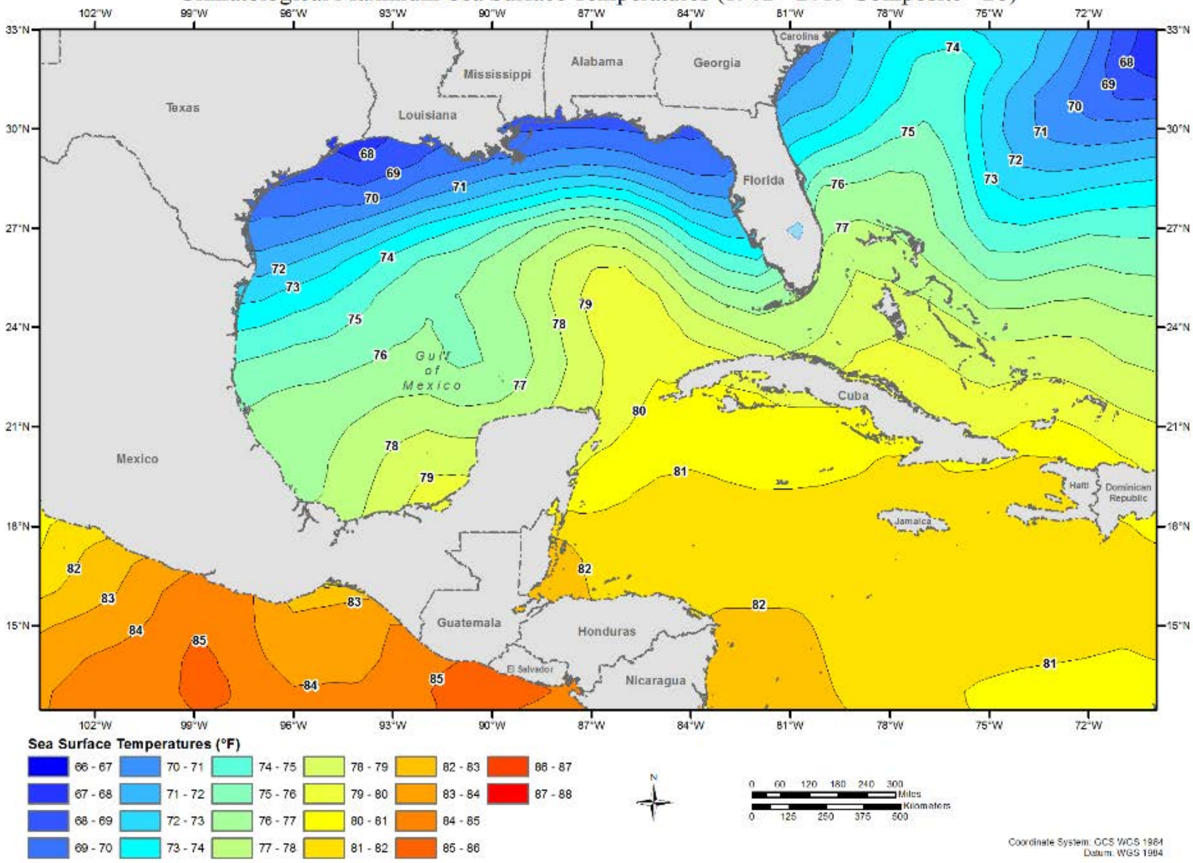
## February

## Climatological Maximum Sea Surface Temperatures (1982 - 2013 Composite +2σ)



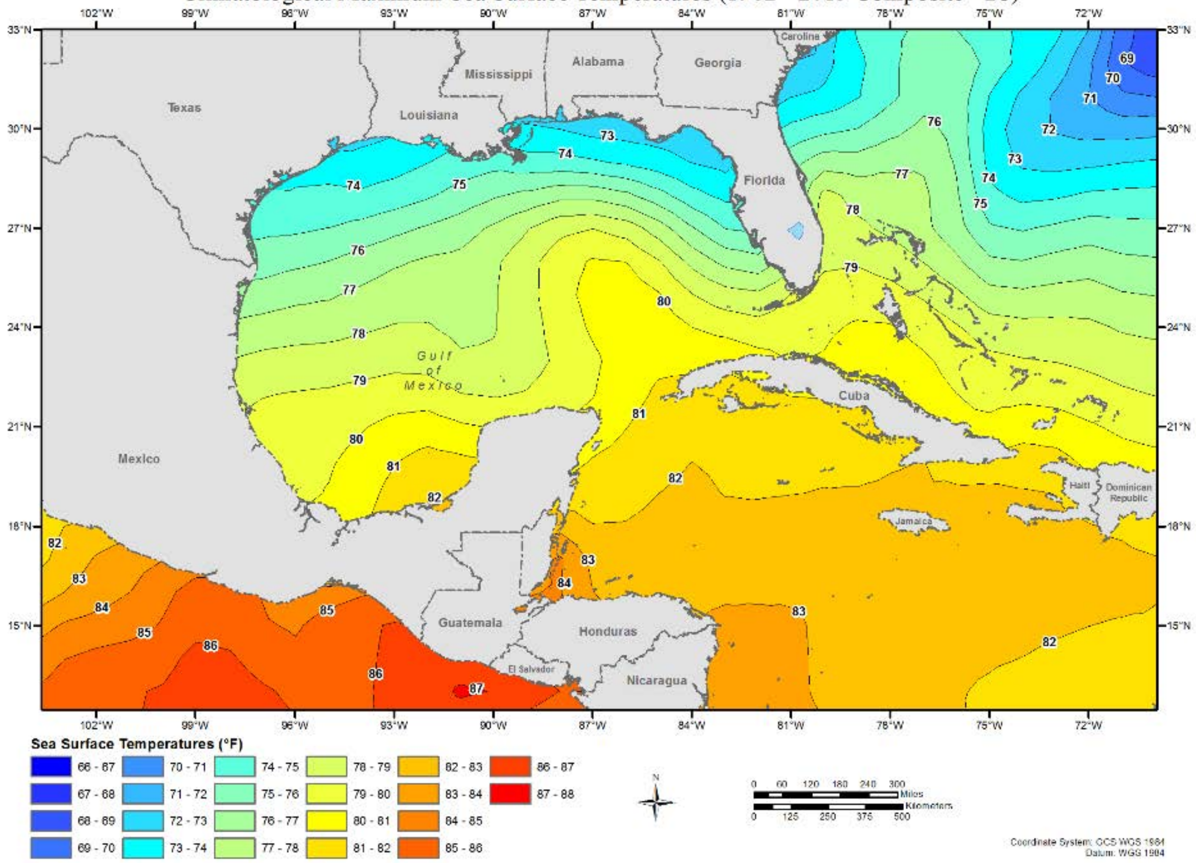
**March**

Climatological Maximum Sea Surface Temperatures (1982 - 2013 Composite +2σ)

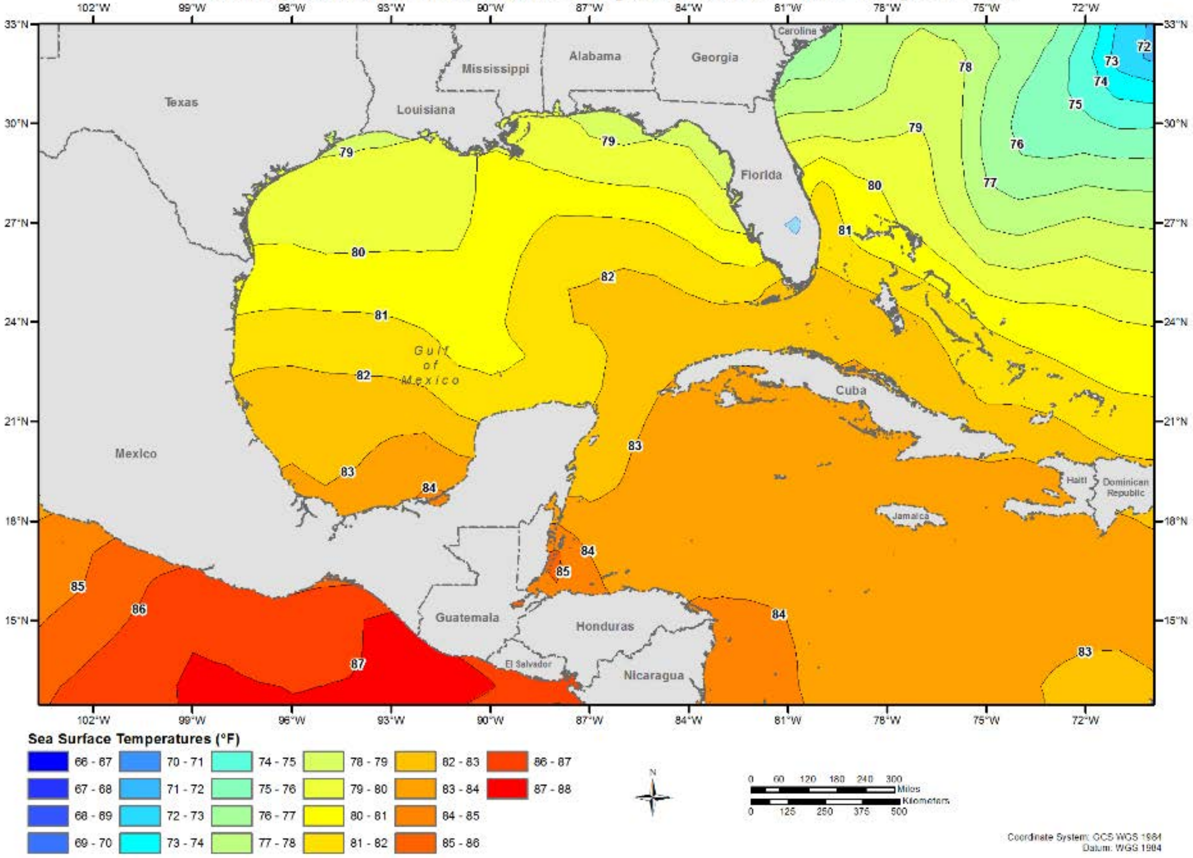


## April

Climatological Maximum Sea Surface Temperatures (1982 - 2013 Composite +2σ)

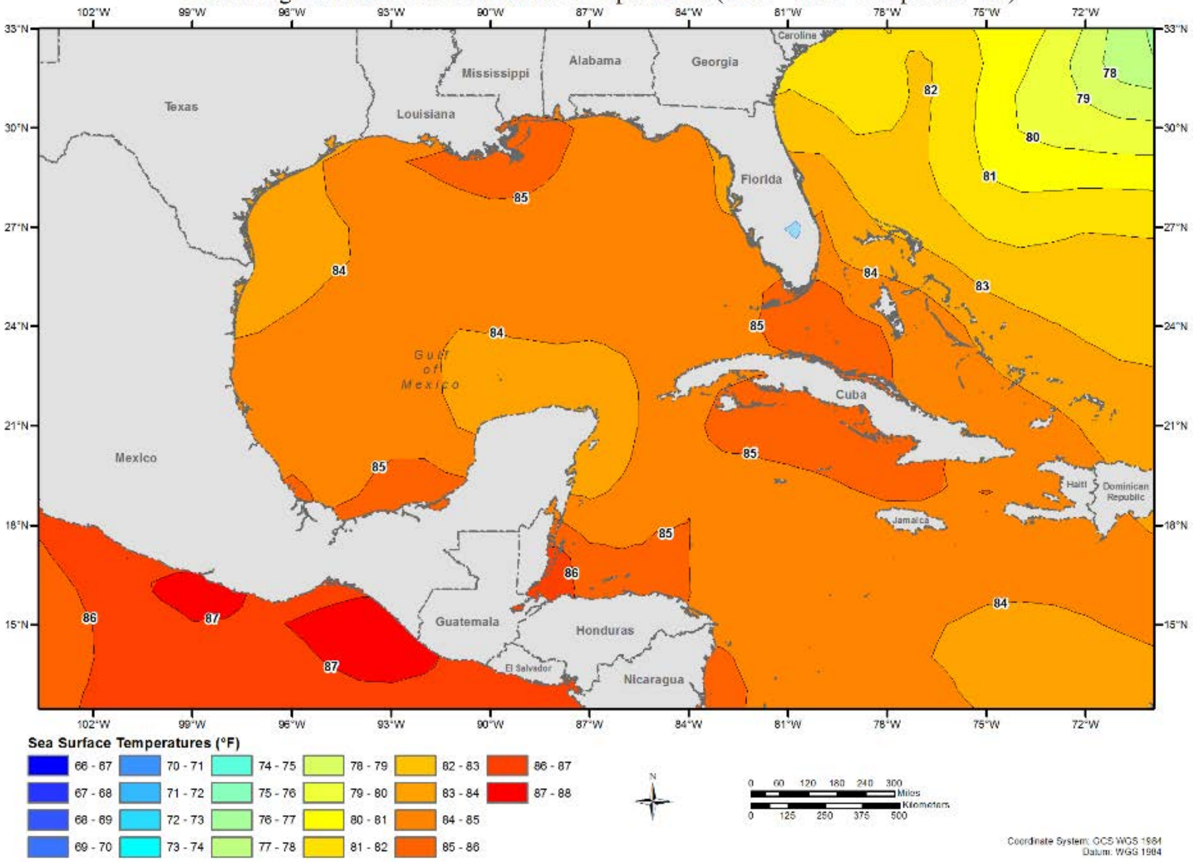


## May
Climatological Maximum Sea Surface Temperatures (1982 - 2013 Composite +2σ)

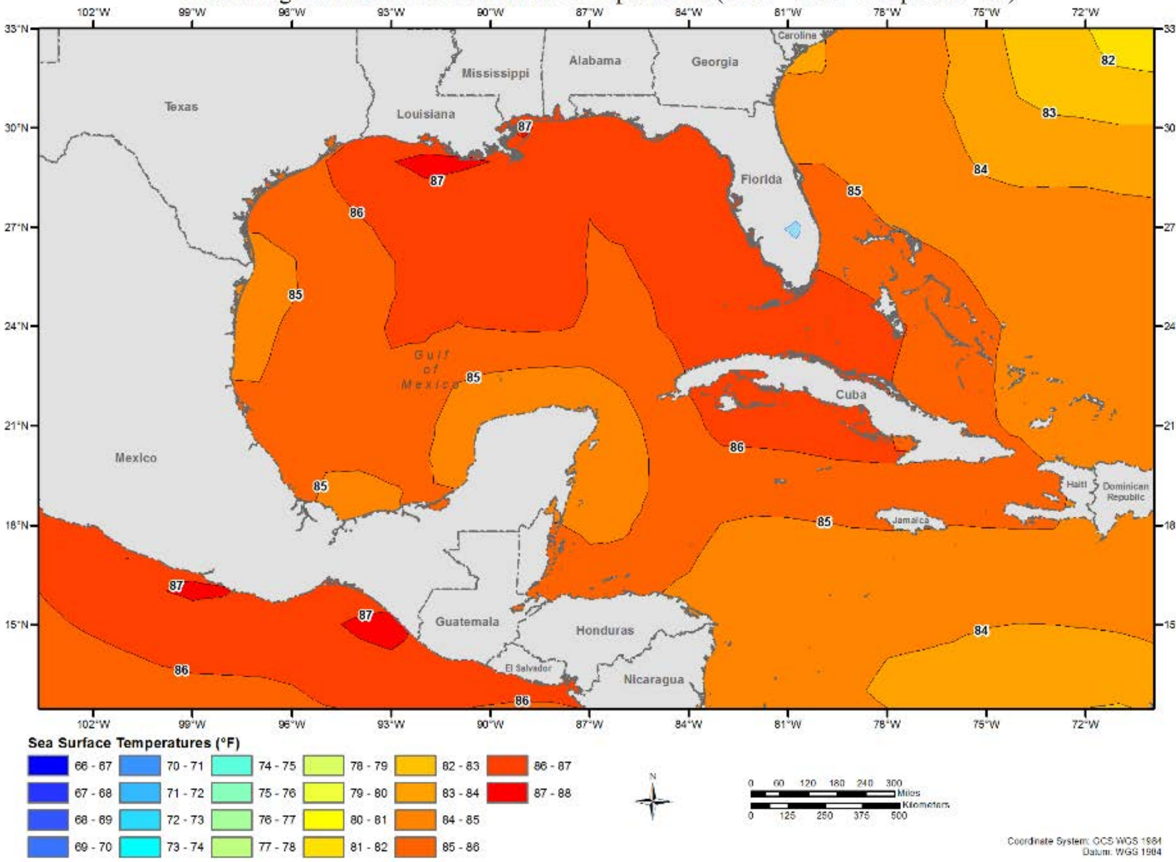


## June

Climatological Maximum Sea Surface Temperatures (1982 - 2013 Composite +2σ)

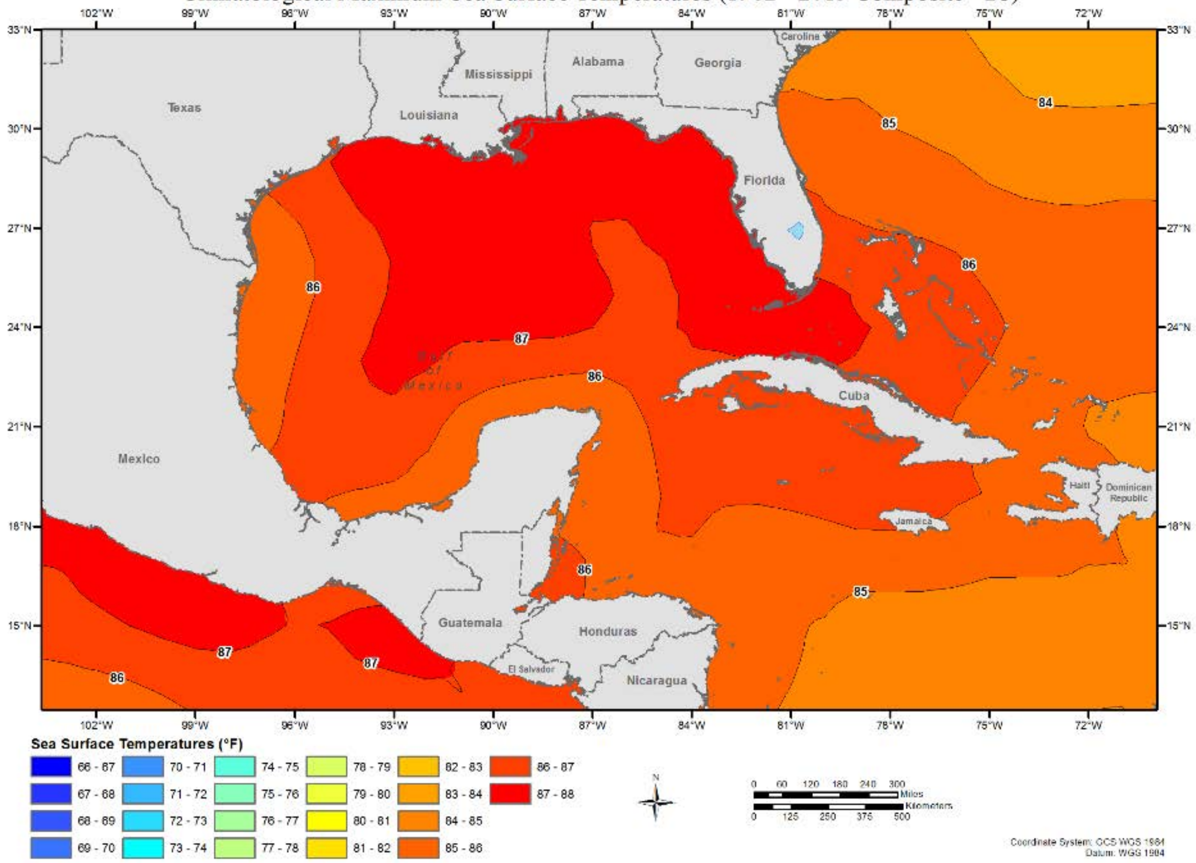


## July
## Climatological Maximum Sea Surface Temperatures (1982 - 2013 Composite +2σ)



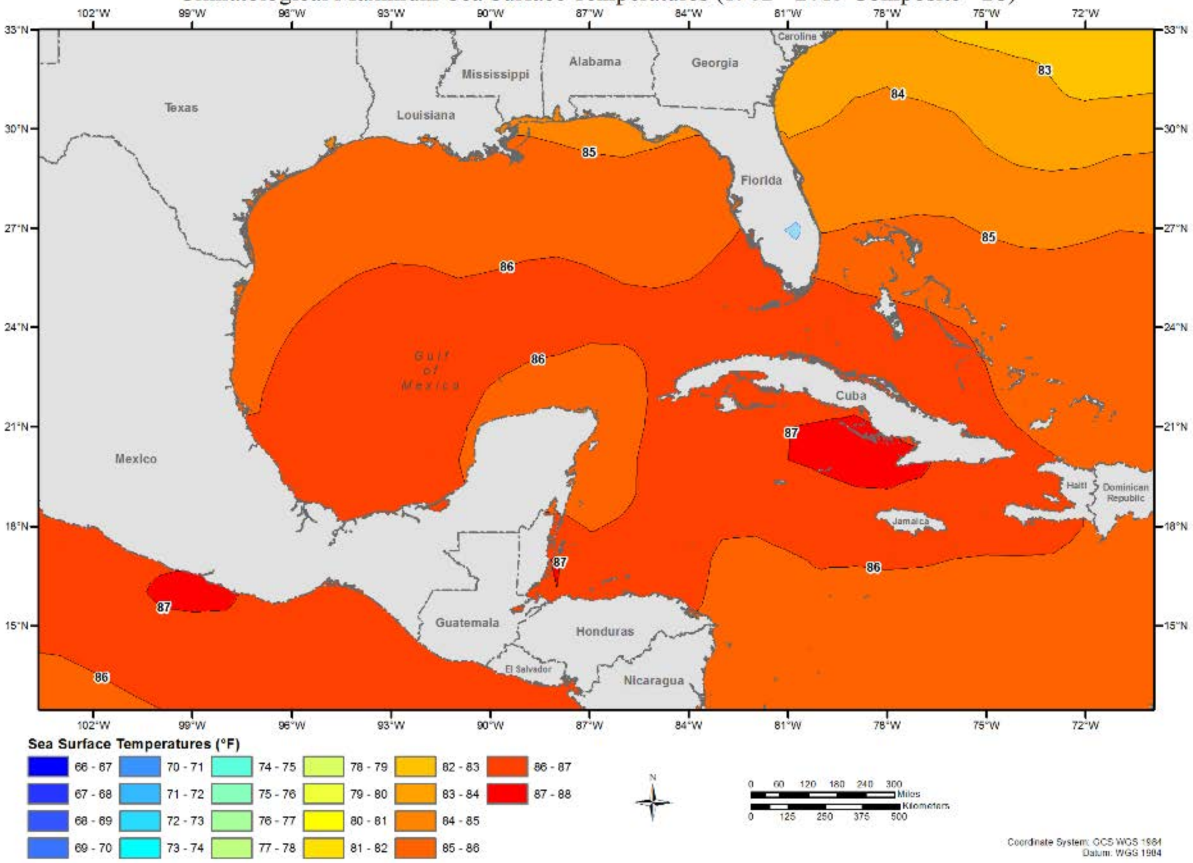
## August

### Climatological Maximum Sea Surface Temperatures (1982 - 2013 Composite +2σ)



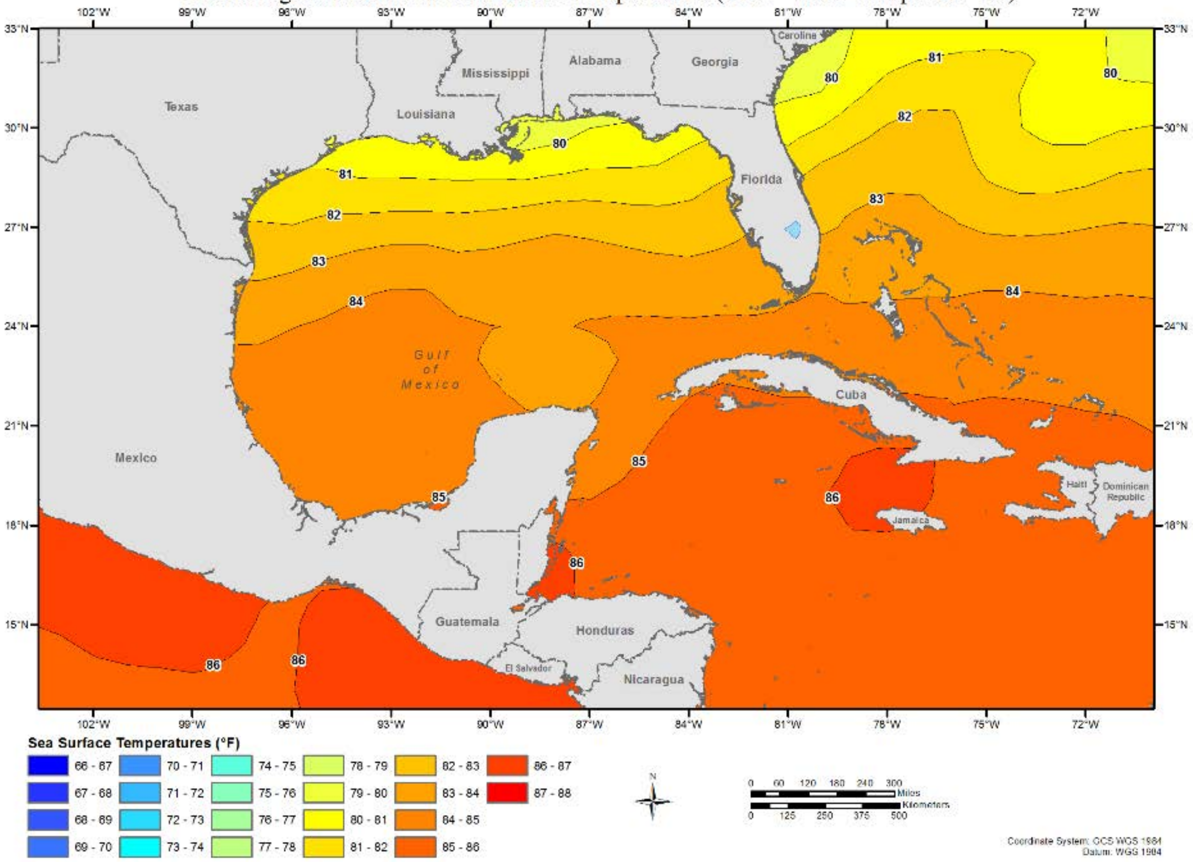
## September

Climatological Maximum Sea Surface Temperatures (1982 - 2013 Composite +2σ)



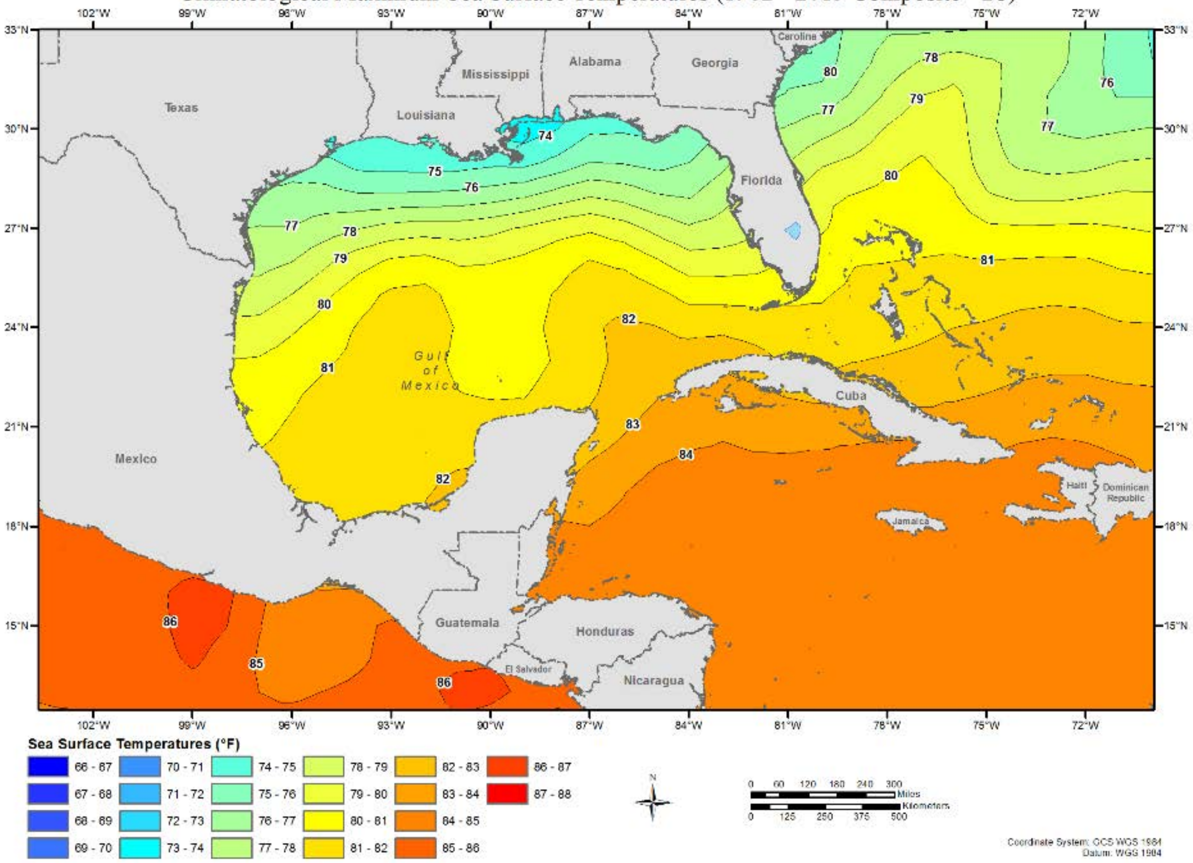
## October

Climatological Maximum Sea Surface Temperatures (1982 - 2013 Composite +2σ)



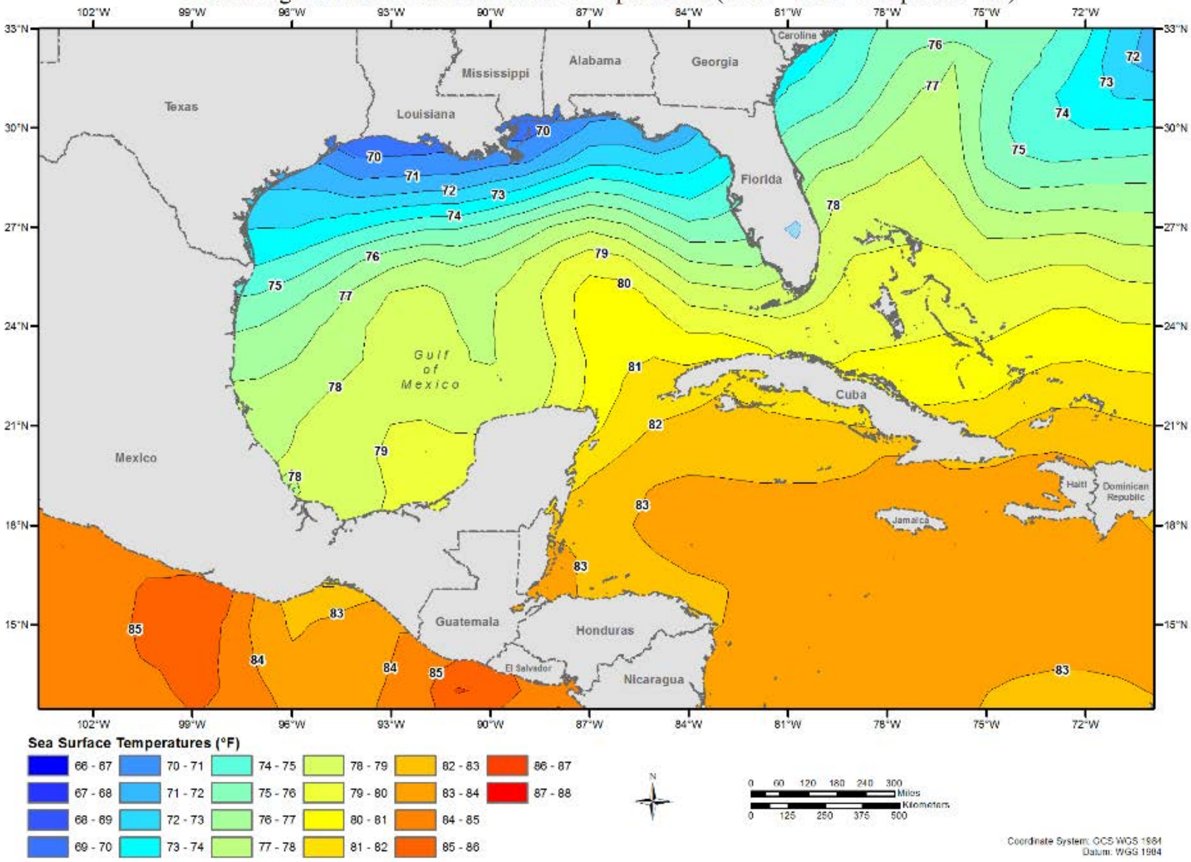
## November

Climatological Maximum Sea Surface Temperatures (1982 - 2013 Composite +2σ)



## December

Climatological Maximum Sea Surface Temperatures (1982 - 2013 Composite +2σ)



# Appendix E
## Storm Precipitation Analysis System (SPAS) Description

# Introduction

The Storm Precipitation Analysis System (SPAS) is grounded on years of scientific research with a demonstrated reliability in hundreds of post-storm precipitation analyses. It has evolved into a trusted hydrometeorological tool that provides accurate precipitation data at a high spatial and temporal resolution for use in a variety of sensitive hydrologic applications (Faulkner et al., 2004, Tomlinson et al., 2003-2012). Applied Weather Associates, LLC and METSTAT, Inc. initially developed SPAS in 2002 for use in producing Depth-Area-Duration values for Probable Maximum Precipitation (PMP) analyses. SPAS utilizes precipitation gauge data, basemaps and radar data (when available) to produce gridded precipitation at time intervals as short as 5 minutes, at spatial scales as fine as 1 km$^2$ and in a variety of customizable formats. To date (March 2015 SPAS has been used to analyze over 500 storm centers across all types of terrain, among highly varied meteorological settings and some occurring over 100-years ago.

SPAS output has many applications including, but not limited to: hydrologic model calibration/validation, flood event reconstruction, storm water runoff analysis, forensic cases and PMP studies. Detailed SPAS-computed precipitation data allow hydrologists to accurately model runoff from basins, particularly when the precipitation is unevenly distributed over the drainage basin or when rain gauge data are limited or not available. The increased spatial and temporal accuracy of precipitation estimates has eliminated the need for commonly made assumptions about precipitation characteristics (such as uniform precipitation over a watershed), thereby greatly improving the precision and reliability of hydrologic analyses.

To instill consistency in SPAS analyses, many of the core methods have remained consistent from the beginning. However, SPAS is constantly evolving and improving through new scientific advancements and as new data and improvements are incorporated. This write-up describes the current inner-workings of SPAS, but the reader should realize SPAS can be customized on a case-by-case basis to account for special circumstances; these adaptations are documented and included in the deliverables. The over-arching goal of SPAS is to combine the strengths of rain gauge data and radar data (when available) to provide sound, reliable and accurate spatial precipitation data.

Hourly precipitation observations are generally limited to a small number of locations, with many basins lacking observational precipitation data entirely. However, Next Generation Radar (NEXRAD) data provide valuable spatial and temporal information over data-sparse basins, which have historically lacked reliability for determining precipitation rates and reliable quantitative precipitation estimates (QPE). The improved reliability in SPAS is made possible by hourly calibration of the NEXRAD radar-precipitation relationship, combined with local hourly bias adjustments to force consistency between the final result and “ground truth” precipitation measurements. If NEXRAD radar data are available (generally for storm events since the mid-1990s), precipitation accumulation at temporal scales as frequent as 5-minutes can be analyzed. If no NEXRAD data are available, then precipitation data are analyzed in hourly increments. A summary of the general SPAS processes is shown in flow chart in Figure E.1.

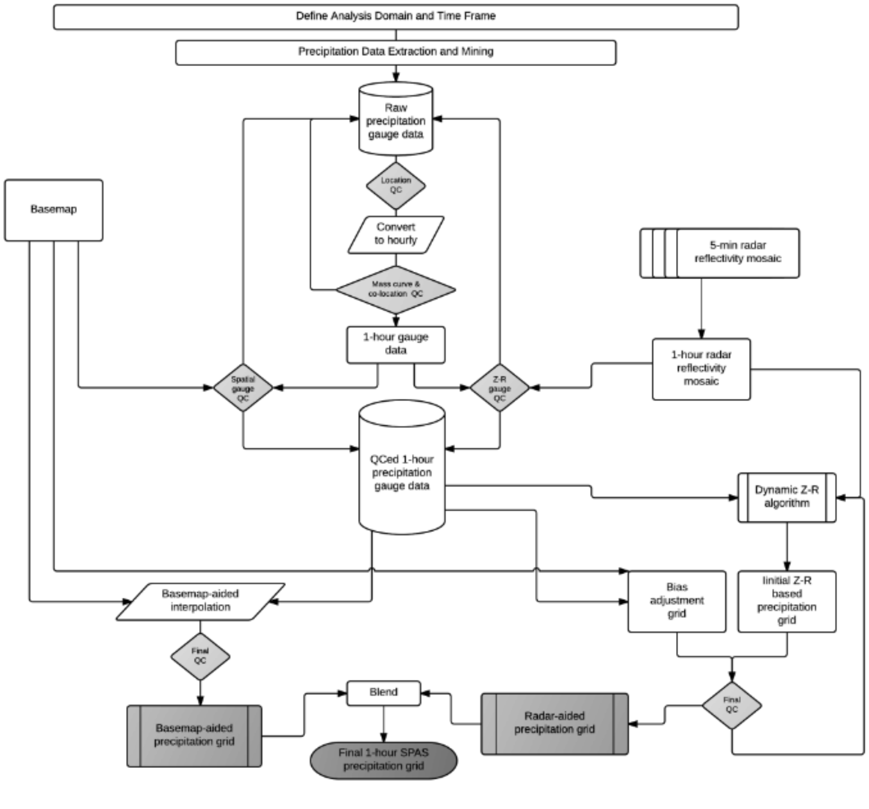

Define Analysis Domain and Time Frame
Precipitation Data Extraction and Mining
Raw precipitation gauge data
Basemap
Location QC
Convert to hourly
5-min radar reflectivity mosaic
Mass curve & co-location QC
1-hour gauge data
1-hour radar reflectivity mosaic
Spatial gauge QC
Z-R gauge QC
QCed 1-hour precipitation gauge data
Dynamic Z-R algorithm
Basemap-aided interpolation
Bias adjustment grid
Iinitial Z-R based precipitation grid
Final QC
Blend
Final QC
Basemap-aided precipitation grid
Final 1-hour SPAS precipitation grid
Radar-aided precipitation grid

**Figure E.1: SPAS flow chart**

# Setup

Prior to a SPAS analysis, careful definition of the storm analysis domain and time frame to be analyzed is established. Several considerations are made to ensure the domain (longitude-latitude box) and time frame are sufficient for the given application.

## SPAS Analysis Domain

For PMP applications it is important to establish an analysis domain that completely encompasses a storm center, meanwhile hydrologic modeling applications are more concerned about a specific basin, watershed or catchment. If radar data are available, then it is also important to establish an area large enough to encompass enough stations (minimum of ~30) to adequately derive reliable radar-precipitation intensity relationships (discussed later). The domain is defined by evaluating existing documentation on the storm as well as plotting and evaluating initial precipitation gauge data on a map. The analysis domain is defined to include as many hourly recording gauges as possible given their importance in timing. The domain must include enough of a buffer to accurately model the nested domain of interest. The domain is defined as a longitude-latitude (upper left and lower right corner) rectangular region.

## SPAS Analysis Time Frame

Ideally, the analysis time frame, also referred to as the Storm Precipitation Period (SPP), will extend from a dry period through the target wet period then back into another dry period. This is to ensure that total storm precipitation amounts can be confidently associated with the storm in question and not contaminated by adjacent wet periods. If this is not possible, a reasonable time

period is selected that is bounded by relatively lighter precipitation. The time frame of the hourly data must be sufficient to capture the full range of daily gauge observational periods for the daily observations to be disaggregated into estimated incremental hourly values (discussed later). For example, if a daily gauge takes observations at 8:00 AM, then the hourly data must be available from 8:00 AM the day prior. Given the configuration of SPAS, the minimum SPP is 72 hours and aligns midnight to midnight.
The core precipitation period (CPP) is a sub-set of the SPP and represents the time period with the most precipitation and the greatest number of reporting gauges. The CPP represents the time period of interest and where our confidence in the results is highest.

## Data

The foundation of a SPAS analysis is the "ground truth" precipitation measurements. In fact, the level of effort involved in "data mining" and quality control represent over half of the total level of effort needed to conduct a complete storm analysis. SPAS operates with three primary data sets: precipitation gauge data, a basemap and, if available, radar data. Table E.1 conveys the variety of precipitation gauges usable by SPAS. For each gauge, the following elements are gathered, entered and archived into SPAS database:

- Station ID
- Station name
- Station type (H=hourly, D=Daily, S=Supplemental, etc.)
- Longitude in decimal degrees
- Latitude in decimal degrees
- Elevation in feet above MSL
- Observed precipitation
- Observation times
- Source
- If unofficial, the measurement equipment and/or method is also noted.

Based on the SPP and analysis domain, hourly and daily precipitation gauge data are extracted from our in-house database as well as the Meteorological Assimilation Data Ingest System (MADIS). Our in-house database contains data dating back to the late 1800s, while the MADIS system (described below) contains archived data back to 2002.

### Hourly Precipitation Data

Our hourly precipitation database is largely comprised of data from NCDC TD-3240, but also precipitation data from other mesonets and meteorological networks (e.g. ALERT, Flood Control Districts, etc.) that we have collected and archived as part of previous studies. Meanwhile, MADIS provides data from a large number of networks across the U.S., including NOAA's HADS (Hydrometeorological Automated Data System), numerous mesonets, the Citizen Weather Observers Program (CWOP), departments of transportation, etc. (see http://madis.noaa.gov/mesonet_providers.html for a list of providers). Although our automatic data extraction is fast, cost-effective and efficient, it never captures all of the available precipitation data for a storm event. For this reason, a thorough "data mining" effort is undertaken to acquire all available data from sources such as U.S. Geological Survey (USGS), Remote Automated Weather Stations (RAWS), Community Collaborative Rain, Hail & Snow Network (CoCoRaHS), National Atmospheric Deposition Program (NADP), Clean Air Status and Trends Network (CASTNET), local observer networks, Climate Reference Network (CRN), Global Summary of the Day (GSD) and Soil Climate Analysis Network (SCAN). Unofficial hourly precipitation data are gathered to give guidance on either timing or magnitude in areas otherwise void of precipitation data. The WeatherUnderground and MesoWest, two of the largest weather databases on the Internet, contain a large proportion of official data, but also includes data from unofficial gauges.

**Table E.1: Different precipitation gauge types used by SPAS**

| Precipitation Gauge Type | Description |
|---|---|
| **Hourly** | Hourly gauges with complete, or nearly complete, incremental hourly precipitation data. |
| **Hourly estimated** | Hourly gauges with some estimated hourly values, but otherwise reliable. |
| **Hourly pseudo** | Hourly gauges with reliable temporal precipitation data, but the magnitude is questionable in relation to co-located daily or supplemental gauge. |
| **Daily** | Daily gauge with complete data and known observation times. |
| **Daily estimated** | Daily gauges with some or all estimated data. |
| **Supplemental** | Gauges with unknown or irregular observation times, but reliable total storm precipitation data. (E.g. public reports, storms reports, “Bucket surveys”, etc.) |
| **Supplemental estimated** | Gauges with estimated total storm precipitation values based on other information (e.g. newspaper articles, stream flow discharge, inferences from nearby gauges, pre-existing total storm isohyetal maps, etc.) |

## Daily Precipitation Data

Our daily database is largely based on NCDC’s TD-3206 (pre-1948) and TD-3200 (1948 through present) as well as SNOTEL data from NRCS. Since the late 1990s, the CoCoRaHS network of more than 15,000 observers in the U.S. has become a very important daily precipitation source. Other daily data are gathered from similar, but smaller gauge networks, for instance the High Spatial Density Precipitation Network in Minnesota.

As part of the daily data extraction process, the time of observation accompanies each measured precipitation value. Accurate observation times are necessary for SPAS to disaggregate the daily precipitation into estimated incremental values (discussed later). Knowing the observation time also allows SPAS to maintain precipitation amounts within given time bounds, thereby retaining known precipitation intensities. Given the importance of observation times, efforts are taken to insure the observation times are accurate. Hardcopy reports of “Climatological Data,” scanned observational forms (available on-line from the NCDC) and/or gauge metadata forms have proven to be valuable and accurate resources for validating observation times. Furthermore, erroneous observation times are identified in the mass-curve quality-control procedure (discussed later) and can be corrected at that point in the process.

## Supplemental Precipitation Gauge Data

For gauges with unknown or irregular observation times, the gauge is considered a “supplemental” gauge. A supplemental gauge can either be added to the storm database with a storm total and the associated SPP as the temporal bounds or as a gauge with the known, but irregular observation times and associated precipitation amounts. For instance, if all that is known is 3 inches fell between 0800-0900, then that information can be entered. Gauges or reports with nothing more than a storm total are often abundant, but to use them, it is important the precipitation is only from the storm period in question. Therefore, it is ideal to have the analysis time frame bounded by dry periods.

Perhaps the most important source of data, if available, is from "bucket surveys," which provide comprehensive lists of precipitation measurements collected during a post-storm field exercise. Although some bucket survey amounts are not from conventional precipitation gauges, they provide important information, especially in areas lacking data. Particularly for PMP-storm analysis applications, it is customary to accept extreme, but valid non-standard precipitation values (such as bottles and other open containers that catch rainfall) to capture the highest precipitation values.

## Basemap

"Basemaps" are independent grids of spatially distributed weather or climate variables that are used to govern the spatial patterns of the hourly precipitation. The basemap also governs the spatial resolution of the final SPAS grids, unless radar data are available/used to govern the spatial resolution. Note that a base map is not required as the hourly precipitation patterns can be based on station characteristics and an inverse distance weighting technique (discussed later). Basemaps in complex terrain are often based on the PRISM mean monthly precipitation (Figure E.2a) or Hydrometeorological Design Studies Center precipitation frequency grids (Figure E.2b) given they resolve orographic enhancement areas and micro-climates at a spatial resolution of 30-seconds (about 800 m). Basemaps of this nature in flat terrain are not as effective given the small terrain forced precipitation gradients. Therefore, basemaps for SPAS analyses in flat terrain are often developed from pre-existing (hand-drawn) isohyetal patterns (Figure E.2c), composite radar imagery or a blend of both.

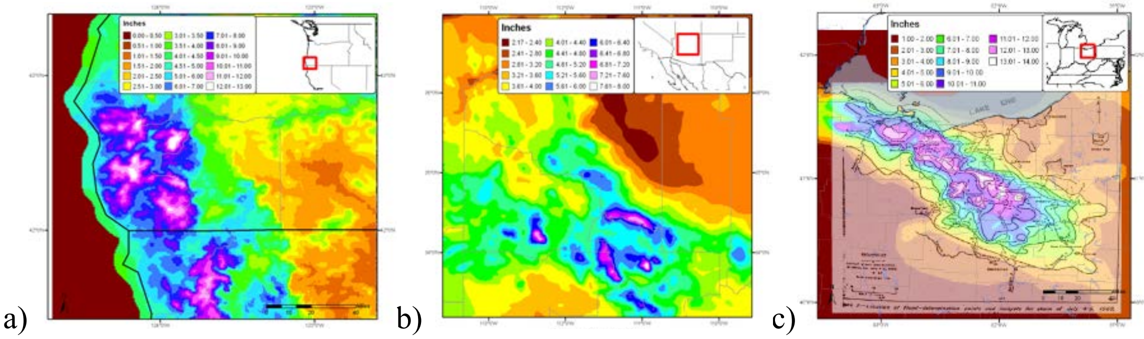
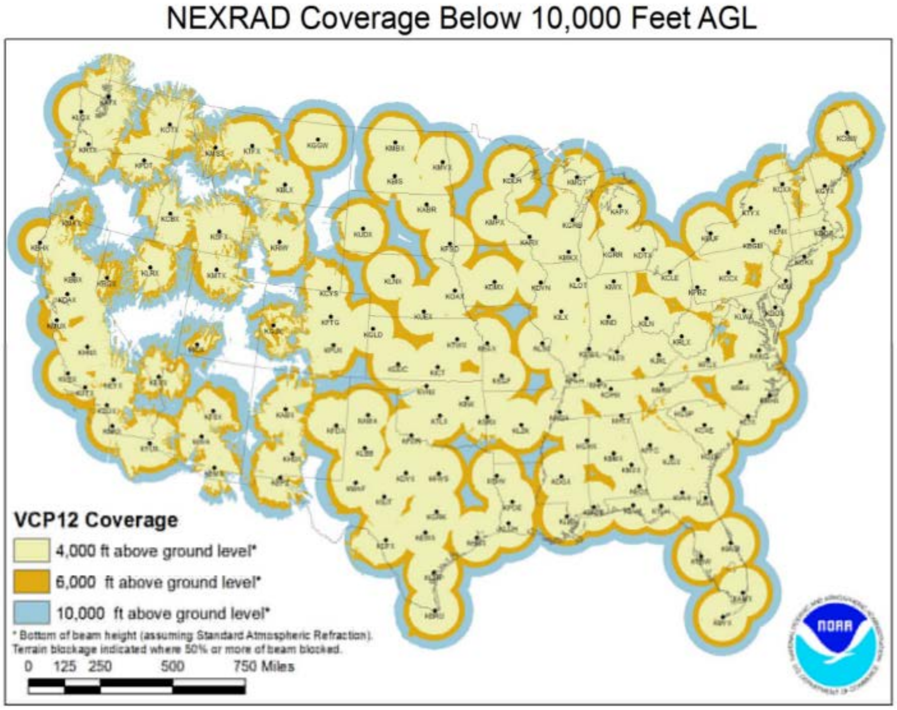


**Figure E.2: Sample SPAS "basemaps:" (a) A pre-existing (USGS) isohyetal pattern across flat terrain (SPAS #1209), (b) PRISM mean monthly (October) precipitation (SPAS #1192) and (c) A 100-year 24-hour precipitation grid from NOAA Atlas 14 (SPAS #1138)**

## Radar Data

For storms occurring since approximately the mid-1990s, weather radar data are available to supplement the SPAS analysis. A fundamental requirement for high quality radar-estimated precipitation is a high quality radar mosaic, which is a seamless collection of concurrent weather radar data from individual radar sites, however in some cases a single radar is sufficient (i.e. for a small area size storm event such as a thunderstorm). Weather radar data have been in use by meteorologists since the 1960s to estimate precipitation depths, but it was not until the early 1990s that new, more accurate NEXRAD Doppler radar (WSR88D) was placed into service across the United States. Currently, efforts are underway to convert the WSR88D radars to dual polarization (DualPol) radar. Today, NEXRAD radar coverage of the contiguous United States is comprised of 159 operational sites and there are 30 in Canada. Each U.S. radar covers an approximate 285 mile (460 km) radial extent while Canadian radars have approximately 256 km (138 nautical miles) radial extent over which their radar can detect precipitation (see Figure E.3). The primary vendor of NEXRAD weather radar data for SPAS is Weather Decision Technologies, Inc. (WDT), who accesses, mosaics, archives and quality-controls NEXRAD radar data from NOAA and Environment Canada. SPAS utilizes Level II NEXRAD radar reflectivity data in units of dBZ, available every 5-minutes in the U.S. and 10-minutes in Canada.


NEXRAD Coverage Below 10,000 Feet AGL

VCP12 Coverage
4,000 ft above ground level*
6,000 ft above ground level*
10,000 ft above ground level*
* Bottom of beam height (assuming Standard Atmospheric Refraction).
Terrain blockage indicated where 50% or more of beam blocked.
0 125 250 500 750 Miles
NOAA

**Figure E.3: U.S. radar locations and their radial extents of coverage below 10,000 feet above ground level (AGL). Each U.S. radar covers an approximate 285 mile radial extent over which the radar can detect precipitation.**

The WDT and National Severe Storms Lab (NSSL) Radar Data Quality Control Algorithm (RDQC) removes non-precipitation artifacts from base Level–II radar data and remaps the data from polar coordinates to a Cartesian (latitude/longitude) grid. Non-precipitation artifacts include ground clutter, bright banding, sea clutter, anomalous propagation, sun strobes, clear air returns, chaff, biological targets, and electronic interference and hardware test patterns. The RDQC algorithm uses sophisticated data processing and a Quality Control Neural Network (QCNN) to delineate the precipitation echoes caused by radar artifacts (Lakshmanan and Valente 2004). Beam blockages due to terrain are mitigated by using 30 meter DEM data to compute and then discard data from a radar beam that clears the ground by less than 50 meters and incurs more than 50% power blockage. A clear-air echo removal scheme is applied to radars in clear-air mode when there is no precipitation reported from observation gauges within the vicinity of the radar. In areas of radar coverage overlap, a distance weighting scheme is applied to assign reflectivity to each grid cell, for multiple vertical levels. This scheme is applied to data from the nearest radar that is unblocked by terrain.

Once data from individual radars have passed through the RDQC, they are merged to create a seamless mosaic for the United States and southern Canada as shown in Figure E.4. A multi-sensor quality control can be applied by post-processing the mosaic to remove any remaining "false echoes." This technique uses observations of infra-red cloud top temperatures by GOES satellite and surface temperature to create a precipitation/no-precipitation mask. Figure E.4(b) shows the impact of WDT's quality control measures. Upon completing all QC, WDT converts the radar data from its native polar coordinate projection (1 degree x 1.0 km) into a longitude-latitude Cartesian grid (based on the WGS84 datum), at a spatial resolution of ~1/3$^{rd}$mi$^2$ for processing in SPAS.

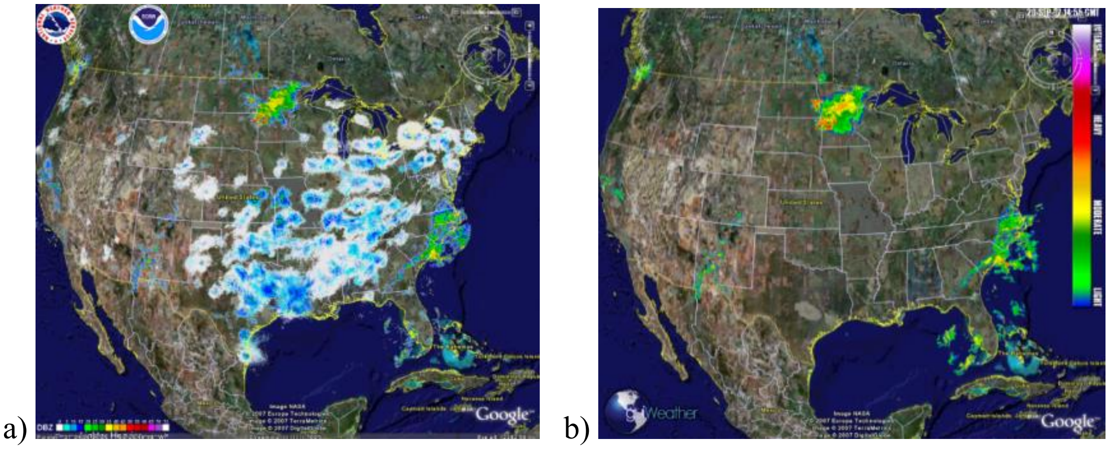


**Figure E.4: (a) Level-II radar mosaic of CONUS radar with no quality control, (b) WDT quality controlled Level-II radar mosaic**

SPAS conducts further QC on the radar mosaic by infilling areas contaminated by beam blockages. Beam blocked areas are objectively determined by evaluating total storm reflectivity grid which naturally amplifies areas of the SPAS analysis domain suffering from beam blockage as shown in Figure E.5.

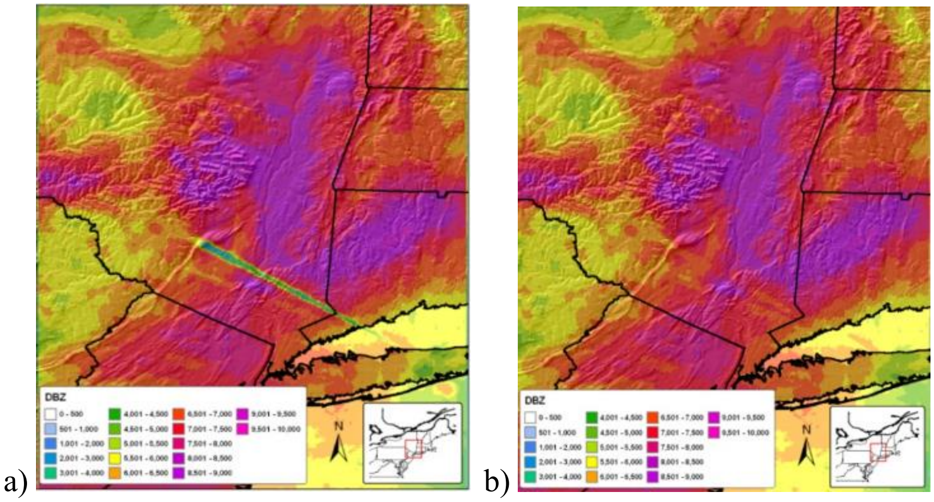


**Figure E.5: Illustration of SPAS-beam blockage infilling where (a) is raw, blocked radar and (b) is filled for a 42-hour storm event**

## Methodology

### Daily and Supplemental Precipitation to Hourly

To obtain one hour temporal resolutions and utilize all gauge data, it is necessary to disaggregate daily and supplemental precipitation observations into estimated hourly amounts. This process has traditionally been accomplished by distributing (temporally) the precipitation at each daily/supplemental gauge in accordance to a single nearby hourly gauge (Thiessen polygon approach). However, this may introduce biases and not correctly represent hourly precipitation at daily/supplemental gauges situated in-between hourly gauges. Instead, SPAS uses a spatial approach by which the estimated hourly precipitation at each daily and supplemental gauge is governed by a distance weighted algorithm of all nearby true hourly gauges.

To disaggregate (i.e. distribute) daily/supplemental gauge data into estimate hourly values, the true hourly gauge data are first evaluated and quality controlled using synoptic maps, nearby gauges, orographic effects, gauge history and other documentation on the storm. Any problems with the hourly data are resolved, and when possible/necessary accumulated hourly values are distributed. If an hourly value is missing, the analyst can choose to either estimate it or leave it missing for SPAS to estimate later based on nearby hourly gauges. At this point in the process, pseudo (hourly) gauges can be added to represent precipitation timing in topographically complex locations, areas with limited/no hourly data or to capture localized convention. Hourly Pseudo stations add additional detail on the timing of rainfall, either from COOP forms, radar reflectivity timing, and/or bucket survey reports with time increments. Hourly Pseudo stations are used only for the timing surrounding daily and supplemental stations and not for the magnitude. The limitations of Hourly Pseudo stations is that they are based on surrogate information, the quality of the information can be highly questionable (based on source) thus the importance of the station QC procedures are extremely important. To adequately capture the temporal variations of the precipitation, a pseudo hourly gauge is sometimes necessary. A pseudo gauge is created by distributing the precipitation at a co-located daily gauge or by creating a completely new pseudo gauge from other information such as inferences from COOP observation forms, METAR visibility data (if hourly precipitation are not already available), lightning data, satellite data, or radar data. Often radar data are the best/only choice for creating pseudo hourly gauges, but this is done cautiously given the potential differences (over-shooting of the radar beam equating to erroneous precipitation) between radar data and precipitation. In any case, the pseudo hourly gauge is flagged so SPAS only uses it for timing and not magnitude. Care is taken to ensure hourly pseudo gauges represent justifiably important physical and meteorological characteristics before being incorporated into the SPAS database. Although pseudo gauges provide a very important role, their use is kept to a minimum. The importance of insuring the reliability of every hourly gauge cannot be over emphasized. All of the final hourly gauge data, including pseudos, are included in the hourly SPAS precipitation database.

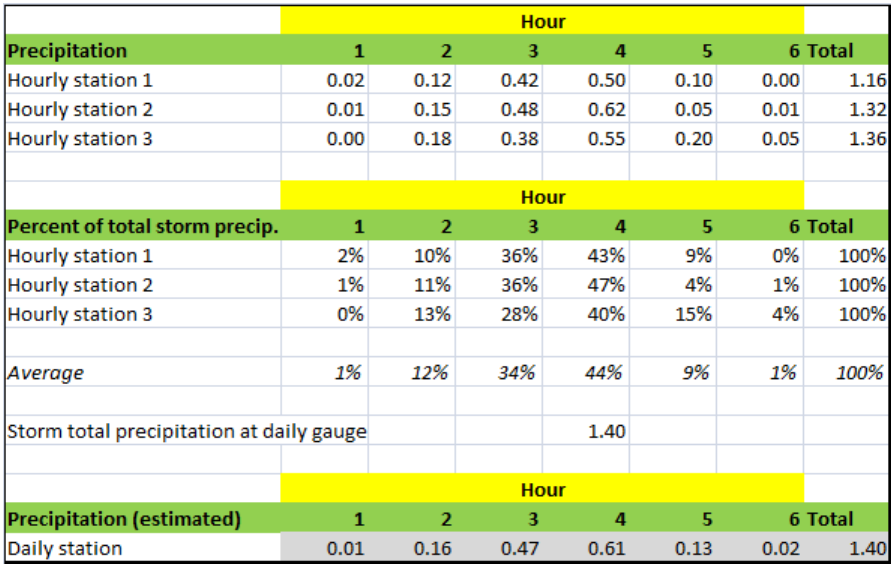
Using the hourly SPAS precipitation database, each hourly precipitation value is converted into a percentage that represents the incremental hourly precipitation divided by the total SPP precipitation. The GIS-ready x-y-z file is constructed for each hour and it includes the latitude (x), longitude(y) and the percent of precipitation (z) for a particular hour. Using the GRASS GIS, an inverse-distance-weighting squared (IDW) interpolation technique is applied to each of the hourly files. The result is a continuous grid with percentage values for the entire analysis domain, keeping the grid cells on which the hourly gauge resides faithful to the observed/actual percentage. Since the percentages typically have a high degree of spatial autocorrelation, the spatial interpolation has skill in determining the percentages between gauges, especially since the percentages are somewhat independent of the precipitation magnitude. The end result is a GIS grid for each hour that represents the percentage of the SPP precipitation that fell during that hour.

After the hourly percentage grids are generated and QC'd for the entire SPP, a program is executed that converts the daily/supplemental gauge data into incremental hourly data. The timing at each of the daily/supplemental gauges is based on (1) the daily/supplemental gauge observation time, (2) daily/supplemental precipitation amount and (3) the series of interpolated hourly percentages extracted from grids (described above).

This procedure is detailed in Figure E.6 below. In this example, a supplemental gauge reported 1.40" of precipitation during the storm event and is located equal distance from the three surrounding hourly recording gauges. The procedure steps are:

Step 1. For each hour, extract the percent of SPP from the hourly gauge-based percentage at the location of the daily/supplemental gauge. In this example, assume these values are the average of all the hourly gauges.

Step 2. Multiply the individual hourly percentages by the total storm precipitation at the daily/supplemental gauge to arrive at estimated hourly precipitation at the daily/supplemental gauge. To make the daily/supplemental accumulated precipitation data faithful to the daily/supplemental observations, it is sometimes necessary to adjust the hourly percentages so they add up to 100% and account for 100% of the daily observed precipitation.



| | Hour | | | | | | |
|---|---|---|---|---|---|---|---|
| **Precipitation** | **1** | **2** | **3** | **4** | **5** | **6** | **Total** |
| Hourly station 1 | 0.02 | 0.12 | 0.42 | 0.50 | 0.10 | 0.00 | 1.16 |
| Hourly station 2 | 0.01 | 0.15 | 0.48 | 0.62 | 0.05 | 0.01 | 1.32 |
| Hourly station 3 | 0.00 | 0.18 | 0.38 | 0.55 | 0.20 | 0.05 | 1.36 |
| | | | | | | | |
| | Hour | | | | | | |
| **Percent of total storm precip.** | **1** | **2** | **3** | **4** | **5** | **6** | **Total** |
| Hourly station 1 | 2% | 10% | 36% | 43% | 9% | 0% | 100% |
| Hourly station 2 | 1% | 11% | 36% | 47% | 4% | 1% | 100% |
| Hourly station 3 | 0% | 13% | 28% | 40% | 15% | 4% | 100% |
| | | | | | | | |
| *Average* | *1%* | *12%* | *34%* | *44%* | *9%* | *1%* | *100%* |
| | | | | | | | |
| Storm total precipitation at daily gauge | | | | 1.40 | | | |
| | | | | | | | |
| | Hour | | | | | | |
| **Precipitation (estimated)** | **1** | **2** | **3** | **4** | **5** | **6** | **Total** |
| Daily station | 0.01 | 0.16 | 0.47 | 0.61 | 0.13 | 0.02 | 1.40 |

**Figure E.6: Example of disaggregation of daily precipitation into estimated hourly precipitation based on three (3) surrounding hourly recording gauges**

In cases where the hourly grids do not indicate any precipitation falling during the daily/supplemental gauge observational period, yet the daily/supplemental gauge reported precipitation, the daily/supplemental total precipitation is evenly distributed throughout the hours that make up the observational period; although this does not happen very often, this solution is consistent with NWS procedures. However, the SPAS analyst is notified of these cases in a comprehensive log file, and in most cases they are resolvable, sometimes with a pseudo hourly gauge.

## Gauge Quality Control

Exhaustive quality control measures are taken throughout the SPAS analysis. Below are a few of the most significant QC measures taken.

### Mass Curve Check

A mass curve-based QC-methodology is used to ensure the timing of precipitation at all gauges is consistent with nearby gauges. SPAS groups each gauge with the nearest four gauges (regardless of type) into a single file. These files are subsequently used in software for graphing and evaluation. Unusual characteristics in the mass curve are investigated and the gauge data corrected, if possible and warranted. See Figure E.7 for an example.

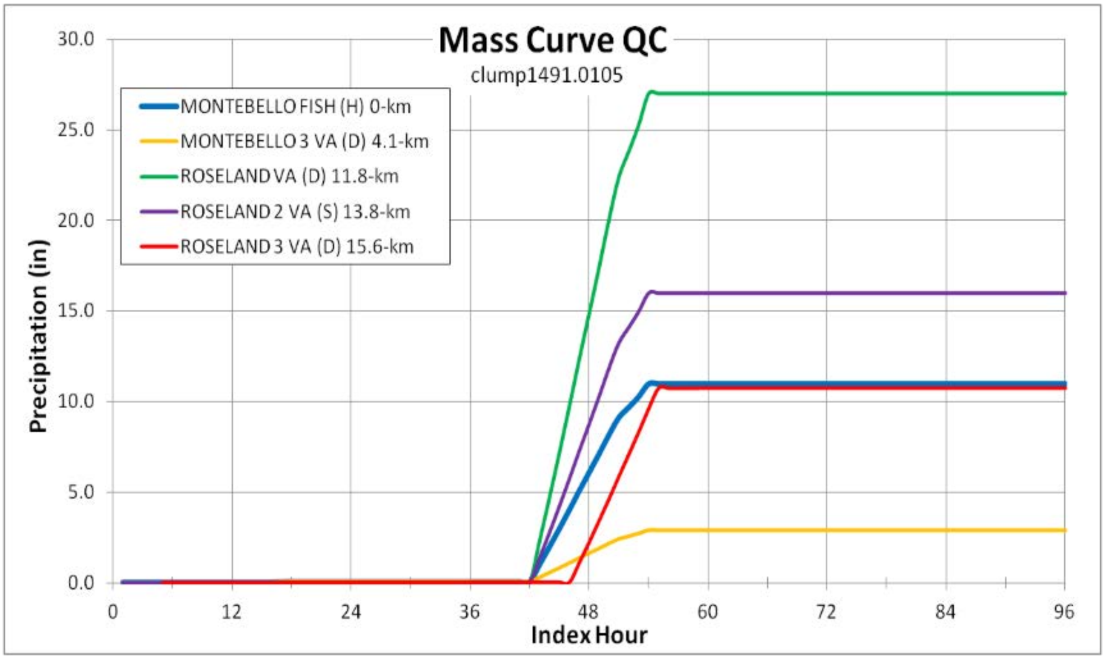


**Figure E.7: Sample mass curve plot depicting a precipitation gauge with an erroneous observation time (red line). X-axis is the SPAS index hour and the y-axis is inches. The statistics in the upper left denote gauge type, and distance from target gauge (in km). In this example, the daily gauge (red line) was found to have an observation error/shift of 6-hours.**

### Gauge Mis-location Check

Although the gauge elevation is not explicitly used in SPAS, it is however used as a means of QC'ing gauge location. Gauge elevations are compared to a high-resolution 15-second DEM to identify gauges with large differences, which may indicate erroneous longitude and/or latitude values.

### Co-located Gauge QC

Care is also taken to establish the most accurate precipitation depths at all co-located gauges. In general, where a co-located gauge pair exists, the highest precipitation is accepted (if deemed accurate). If the hourly gauge reports higher precipitation, then the co-located daily (or supplemental) is removed from the analysis since it would not add anything to the analysis. Often daily (or supplemental) gauges report greater precipitation than a co-located hourly station since hourly tipping bucket gauges tend to suffer from gauge under-catch, particularly during extreme events, due to loss of precipitation during tips. In these cases the daily/supplemental is retained for the magnitude and the hourly used as a pseudo hourly gauge for timing. Large discrepancies between any co-located gauges are investigated and resolved since SPAS can only utilize a single gauge magnitude at each co-located site.

## Spatial Interpolation

At this point the QC'd observed hourly and disaggregated daily/supplemental hourly precipitation data are spatially interpolated into hourly precipitation grids. SPAS has three options for conducting the hourly precipitation interpolation, depending on the terrain and availability of radar data, thereby allowing SPAS to be optimized for any particular storm type or location. Figure E.8 depicts the results of each spatial interpolation methodology based on the same precipitation gauge data.

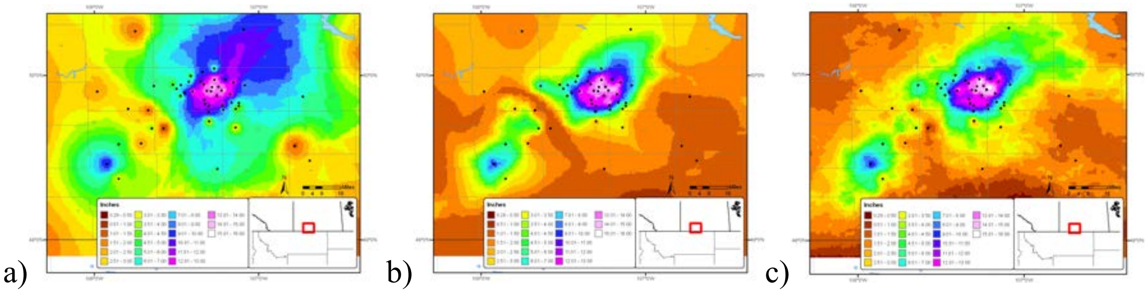


**Figure E.8: Depictions of total storm precipitation based on the three SPAS interpolation methodologies for a storm (SPAS #1177, Vanguard, Canada) across flat terrain: (a) no basemap, (b) basemap-aided and (c) radar**

### Basic Approach

The basic approach interpolates the hourly precipitation point values to a grid using an inverse distance weighting squared GIS algorithm. This is sometimes the best choice for convective storms over flat terrain when radar data are not available, yet high gauge density instills reliable precipitation patterns. This approach is rarely used.

### Basemap Approach

Another option includes use of a basemap, also known as a climatologically-aided interpolation (Hunter 2005). As noted before, the spatial patterns of the basemap govern the interpolation between points of hourly precipitation estimates, while the actual hourly precipitation values govern the magnitude. This approach to interpolating point data across complex terrain is widely used. In fact, it was used extensively by the NWS during their storm analysis era from the 1940s through the 1970s (USACE 1973, Hansen et al., 1988, Corrigan et al., 1999).

In application, the hourly precipitation gauge values are first normalized by the corresponding grid cell value of the basemap before being interpolated. The normalization allows information and knowledge from the basemap to be transferred to the spatial distribution of the hourly precipitation. Using an IDW squared algorithm, the normalized hourly precipitation values are interpolated to a grid. The resulting grid is then multiplied by the basemap grid to produce the hourly precipitation grid. This is repeated each hour of the storm.

## Radar Approach

The coupling of SPAS with NEXRAD provides the most accurate method of spatially and temporally distributing precipitation. To increase the accuracy of the results however, quality-controlled precipitation observations are used for calibrating the radar reflectivity to rain rate relationship (Z-R relationship) each hour instead of assuming a default Z-R relationship. Also, spatial variability in the Z-R relationship is accounted for through local bias corrections (described later). The radar approach involves several steps, each briefly described below. The radar approach cannot operate alone – either the basic or basemap approach must be completed before radar data can be incorporated. The SPAS general code is where the daily and supplemental station are timed to hourly data. Therefore, to get the correct timing of daily and supplemental stations, SPAS general needs to be run. The timed hourly data are used as input into SPAS-NEXRAD to derive the dynamic ZR relationship each hour.

Basemaps are only used to aid in the spatial interpolation. In regards to SPAS-NEXRAD, a basemap is used to interpolate the radar residuals (bias adjustments).

## Z-R Relationship

SPAS derives high quality precipitation estimates by relating quality controlled level–II NEXRAD radar reflectivity radar data with quality-controlled precipitation gauge data to calibrate the Z-R (radar reflectivity, Z, and precipitation, R) relationship. Optimizing the Z-R relationship is essential for capturing temporal changes in the Z-R. Most current radar-derived precipitation techniques rely on a constant relationship between radar reflectivity and precipitation rate for a given storm type (e.g. tropical, convective), vertical structure of reflectivity and/or reflectivity magnitudes. This non-linear relationship is described by the Z-R equation below:

$$Z = A\, R^{b} \qquad (1)$$

Where Z is the radar reflectivity (measured in units of dBZ), R is the precipitation (precipitation) rate (millimeters per hour), A is the “multiplicative coefficient” and b is the “power coefficient”. Both A and b are directly related to the rain drop size distribution (DSD) and rain drop number distribution (DND) within a cloud (Martner and Dubovskiy 2005). The variability in the results of Z versus R is a direct result of differing DSD, DND and air mass characteristics (Dickens 2003). The DSD and DND are determined by complex interactions of microphysical processes that fluctuate regionally, seasonally, daily, hourly, and even within the same cloud. For these reasons, SPAS calculates an optimized Z-R relationship across the analysis domain each hour, based on observed precipitation rates and radar reflectivity (see Figure E.9).

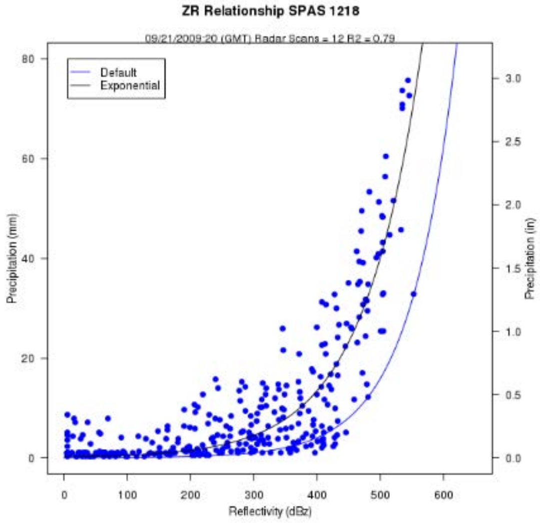
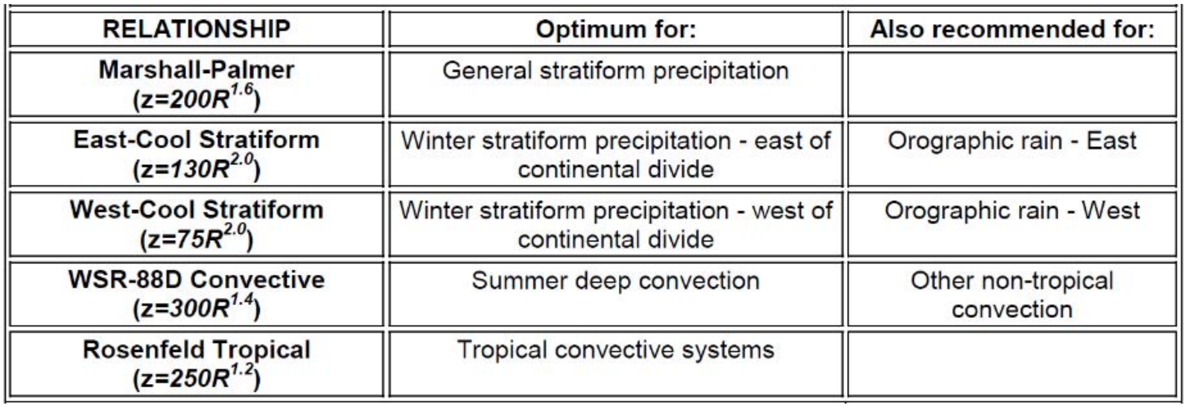


**Figure E.9: Example SPAS (denoted as "Exponential") vs. default Z-R relationship (SPAS #1218, Georgia September 2009)**

The National Weather Service (NWS) utilizes different default Z-R algorithms, depending on the type of precipitation event, to estimate precipitation from NEXRAD radar reflectivity data across the United States (see Figure E.10) (Baeck and Smith 1998 and Hunter 1999). A default Z-R relationship of Z = 300R1.4 is the primary algorithm used throughout the continental U.S. However, it is widely known that this, compared to unadjusted radar-aided estimates of precipitation, suffers from deficiencies that may lead to significant over or under-estimation of precipitation.



| RELATIONSHIP | Optimum for: | Also recommended for: |
|---|---|---|
| Marshall-Palmer<br>($z=200R^{1.6}$) | General stratiform precipitation | |
| East-Cool Stratiform<br>($z=130R^{2.0}$) | Winter stratiform precipitation - east of continental divide | Orographic rain - East |
| West-Cool Stratiform<br>($z=75R^{2.0}$) | Winter stratiform precipitation - west of continental divide | Orographic rain - West |
| WSR-88D Convective<br>($z=300R^{1.4}$) | Summer deep convection | Other non-tropical convection |
| Rosenfeld Tropical<br>($z=250R^{1.2}$) | Tropical convective systems | |

**Figure E.10: Commonly used Z-R algorithms used by the NWS**

Instead of adopting a standard Z-R, SPAS utilizes a least squares fit procedure for optimizing the Z-R relationship each hour of the SPP. The process begins by determining if sufficient (minimum 12) observed hourly precipitation and radar data pairs are available to compute a reliable Z-R. If insufficient (<12) gauge pairs are available, then SPAS adopts the previous hour Z-R relationship, if available, or applies a user-defined default Z-R algorithm. If sufficient data are available, the one hour sum of NEXRAD reflectivity (Z) is related to the 1-hour precipitation at each gauge. A least-squares-fit exponential function using the data points is computed. The resulting best-fit, one hour-based Z-R is subjected to several tests to determine if the Z-R relationship and its resulting precipitation rates are within a certain tolerance based on the R-squared fit measure and difference between the derived and default Z-R precipitation results. Experience has shown the actual Z-R versus the default Z-R can be significantly different (Figure E.11). These Z-R relationships vary by storm type and location. A standard output of all SPAS analyses utilizing NEXRAD includes a file with each hour's adjusted Z-R relationship as calculated through the SPAS program.

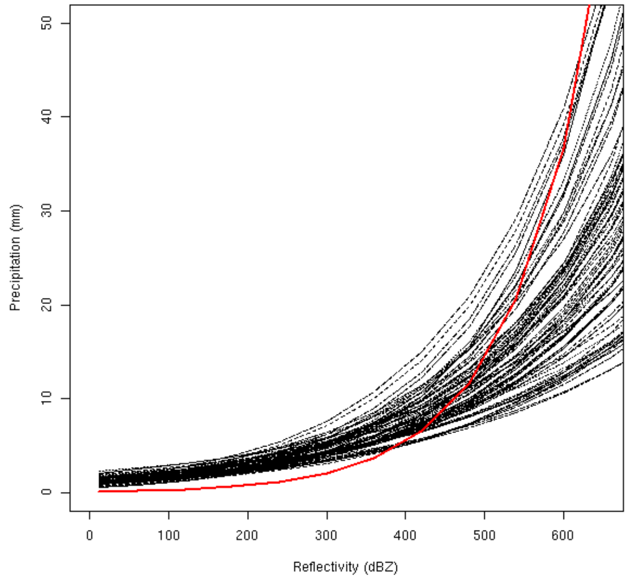


**Figure E.11: Comparison of the SPAS optimized hourly Z-R relationships (black lines) versus a default Z=75R2.0 Z-R relationship (red line) for a period of 99 hours for a storm over southern California.**

## Radar-aided Hourly Precipitation Grids

Once a mathematically optimized hourly Z-R relationship is determined, it is applied to the total hourly Z grid to compute an initial precipitation rate (inches/hour) at each grid cell. To account for spatial differences in the Z-R relationship, SPAS computes residuals, the difference between the initial precipitation analysis (via the Z-R equation) and the actual "ground truth" precipitation (observed – initial analysis), at each gauge. The point residuals, also referred to as local biases, are normalized and interpolated to a residual grid using an inverse distance squared weighting algorithm. A radar-based hourly precipitation grid is created by adding the residual grid to the initial grid; this allows precipitation at the grid cells for which gauges are "on" to be true and faithful to the gauge measurement. The pre-final radar-aided precipitation grid is subject to some final, visual QC checks to ensure the precipitation patterns are consistent with the terrain; these checks are particularly important in areas of complex terrain where even QC'd radar data can be unreliable. The next incremental improvement with SPAS program will come as the NEXRAD radar sites are upgraded to dual-polarimetric capability.

## Radar- and Basemap-Aided Hourly Precipitation Grids

At this stage of the radar approach, a radar- and basemap-aided hourly precipitation grid exists for each hour. At locations with precipitation gauges, the grids are equal, however elsewhere the grids can vary for a number of reasons. For instance, the basemap-aided hourly precipitation grid may depict heavy precipitation in an area of complex terrain, blocked by the radar, whereas the radar-aided hourly precipitation grid may suggest little, if any, precipitation fell in the same area. Similarly, the radar-aided hourly precipitation grid may depict an area of heavy precipitation in flat terrain that the basemap-approach missed since the area of heavy precipitation occurred in an area without gauges. SPAS uses an algorithm to compute the hourly precipitation at each pixel given the two results. Areas that are completely blocked from a radar signal are accounted for with the basemap-aided results (discussed earlier). Precipitation in areas with orographically effective terrain and reliable radar data are governed by a blend of the basemap- and radar-aided precipitation. Elsewhere, the radar-aided precipitation is used exclusively. This blended approach has proven effective for resolving precipitation in complex terrain, yet retaining accurate radar-aided precipitation across areas where radar data are reliable. Figure E.12 illustrates the evolution of final precipitation from radar reflectivity in an area of complex terrain in southern California.

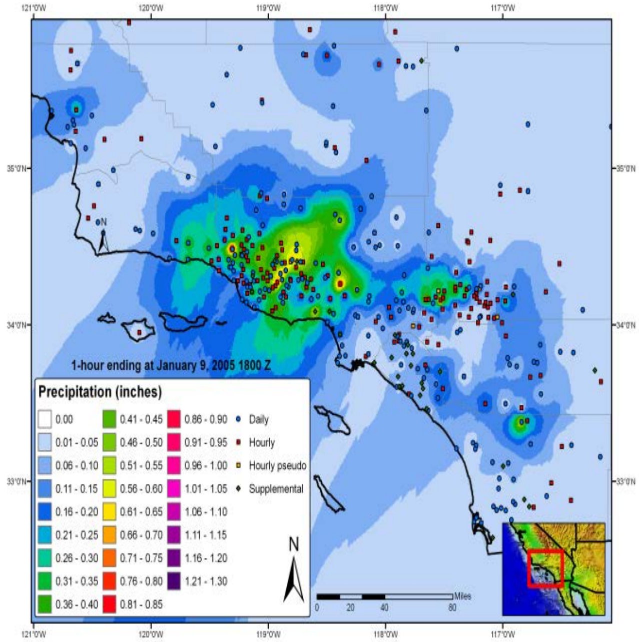


**Figure E.12a: Map depicting 1-hour of precipitation utilizing inverse distance weighting of gauge precipitation for a January 2005 storm in southern California, USA**

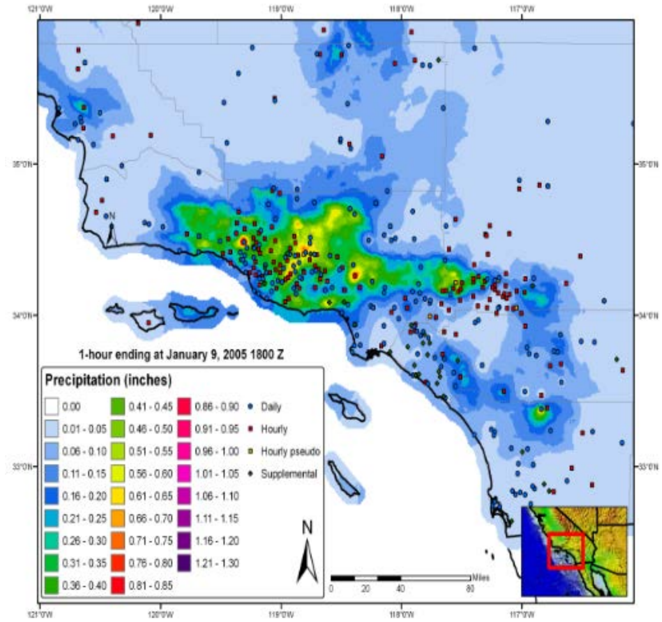


**Figure E.12b: Map depicting 1-hour of precipitation utilizing gauge data together with a climatologically-aided interpolation scheme for a January 2005 storm in southern California, USA**

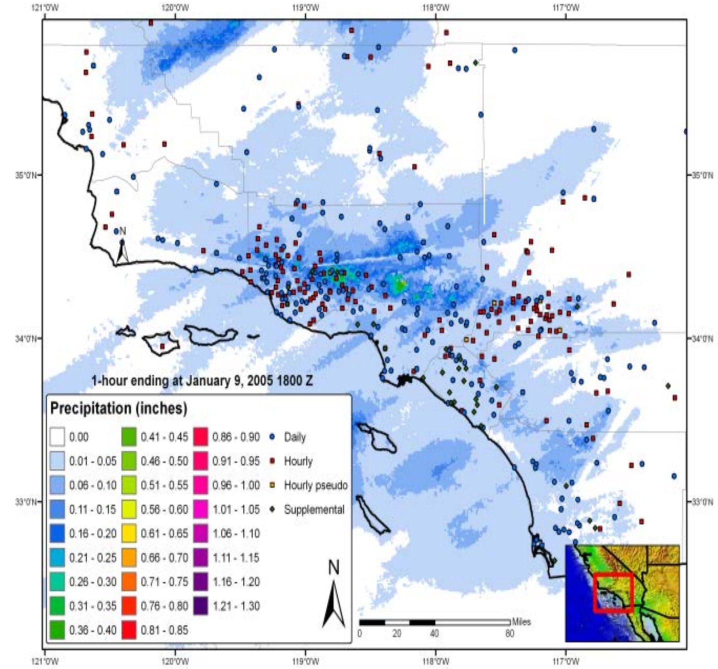


**Figure E.12c: Map depicting 1-hour of precipitation utilizing default Z-R radar-estimated interpolation (no gauge correction) for a January 2005 storm in southern California, USA**

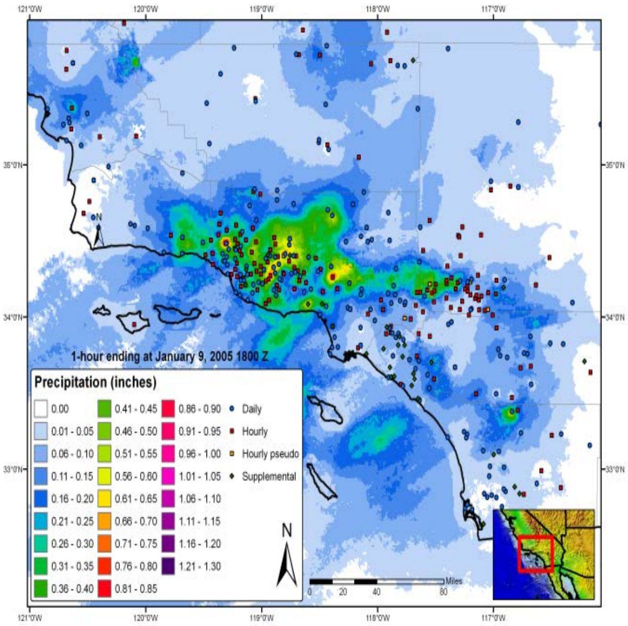


**Figure E.12d: Map depicting 1-hour of precipitation utilizing SPAS precipitation for a January 2005 storm in southern California, USA**

## SPAS versus Gauge Precipitation

Performance measures are computed and evaluated each hour to detect errors and inconsistencies in the analysis. The measures include: hourly Z-R coefficients, observed hourly maximum precipitation, maximum gridded precipitation, hourly bias, hourly mean absolute error (MAE), root mean square error (RMSE), and hourly coefficient of determination ($r^2$).

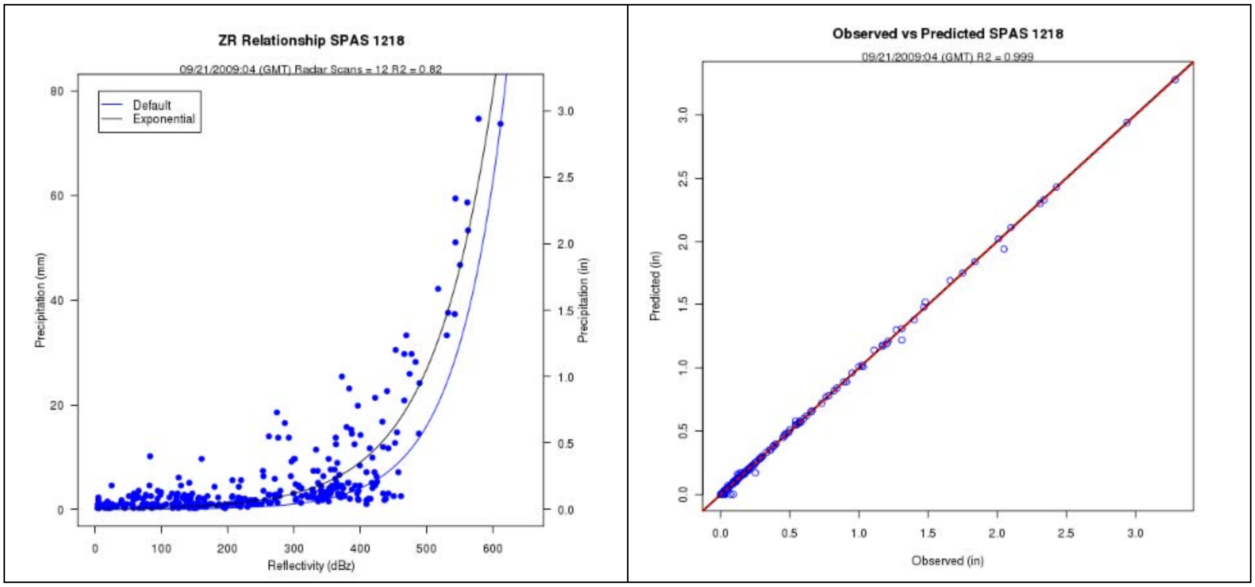


**Figure E.13: Z-R plot (a), where the blue line is the SPAS derived Z-R and the black line is the default Z-R, and the (b) associated observed versus SPAS scatter plot at gauge locations.**

Comparing SPAS-calculated precipitation ($R_{spas}$) to observed point precipitation depths at the gauge locations provides an objective measure of the consistency, accuracy and bias. Generally speaking SPAS is usually within 5% of the observed precipitation (see Figure E.13). Less-than-perfect correlations between SPAS precipitation depths and observed precipitation at gauged locations could be the result of any number of issues, including:

- **Point versus area:** A rain gauge observation represents a much smaller area than the area sampled by the radar. The area that the radar is sampling is approximately 1 $km^2$, whereas a standard rain gauge has an opening 8 inches in diameter, hence it only samples approximately $8.0x10^{-9}$ $km^2$. Furthermore, the radar data represent an average reflectivity (Z) over the grid cell, when in fact the reflectivity can vary across the 1 $km^2$ grid cell. Therefore, comparing a grid cell radar derived precipitation value to a gauge (point) precipitation depth measured may vary.
- **Precipitation gauge under-catch:** Although we consider gauge data "ground truth," we recognize gauges themselves suffer from inaccuracies. Precipitation gauges, shielded and unshielded, inherently underestimate total precipitation due to local airflow, wind under-catch, wetting, and evaporation. The wind under-catch errors are usually around 5% but can be as large as 40% in high winds (Guo et al., 2001, Duchon and Essenberg 2001, Ciach 2003, Tokay et al., 2010). Tipping buckets miss a small amount of precipitation during each tip of the bucket due to the bucket travel and tip time. As precipitation intensities increase, the volumetric loss of precipitation due to tipping tends to increase. Smaller tipping buckets can have higher volumetric losses due to higher tip frequencies, but on the other hand capture higher precision timing.
- **Radar Calibration:** NEXRAD radars calibrate reflectivity every volume scan, using an internally generated test. The test determines changes in internal variables such as beam power and path loss of the receiver signal processor since the last off-line calibration. If this value becomes large, it is likely that there is a radar calibration error that will translate into less reliable precipitation estimates. The calibration test is supposed to maintain a reflectivity precision of 1 dBZ. A 1 dBZ error can result in an error of up to 17% in $R_{spas}$ using the default Z-R relationship $Z=300R^{1.4}$. Higher calibration errors will result in higher $R_{spas}$ errors. However, by performing correlations each hour, the calibration issue is minimized in SPAS.
- **Attenuation:** Attenuation is the reduction in power of the radar beams' energy as it travels from the antenna to the target and back. It is caused by the absorption and the scattering of power from the beam by precipitation. Attenuation can result in errors in Z as large as 1 dBZ especially when the radar beam is sampling a large area of heavy precipitation. In some cases, storm precipitation is so intense (>12 inches/hour) that individual storm cells become "opaque" and the radar beam is totally attenuated. Armed with sufficient gauge data however, SPAS will overcome attenuation issues.
- **Range effects:** The curvature of Earth and radar beam refraction result in the radar beam becoming more elevated above the surface with increasing range. With the increased elevation of the radar beam comes a decrease in Z values due to the radar beam not sampling the main precipitation portion of the cloud (i.e. "over topping" the precipitation and/or cloud altogether). Additionally, as the radar beam gets further from the radar, it naturally samples a larger and larger area, therefore amplifying point versus area differences (described above).
- **Radar Beam Occultation/Ground Clutter:** Radar occultation (beam blockage) results when the radar beam's energy intersects terrain features as depicted in Figure E.14. The result is an increase in radar reflectivity values that can result in higher than normal precipitation estimates. The WDT processing algorithms account for these issues, but SPAS uses GIS spatial interpolation functions to infill areas suffering from poor or no radar coverage.
- **Anomalous Propagation (AP):** AP is false reflectivity echoes produced by unusual rates of refraction in the atmosphere. WDT algorithms remove most of the AP and false echoes, however in extreme cases the air near the ground may be so cold and dense that a radar beam that starts out moving upward is bent all the way down to the ground. This produces erroneously strong echoes at large distances from the radar. Again, equipped with sufficient gauge data, the SPAS bias corrections will overcome AP issues.

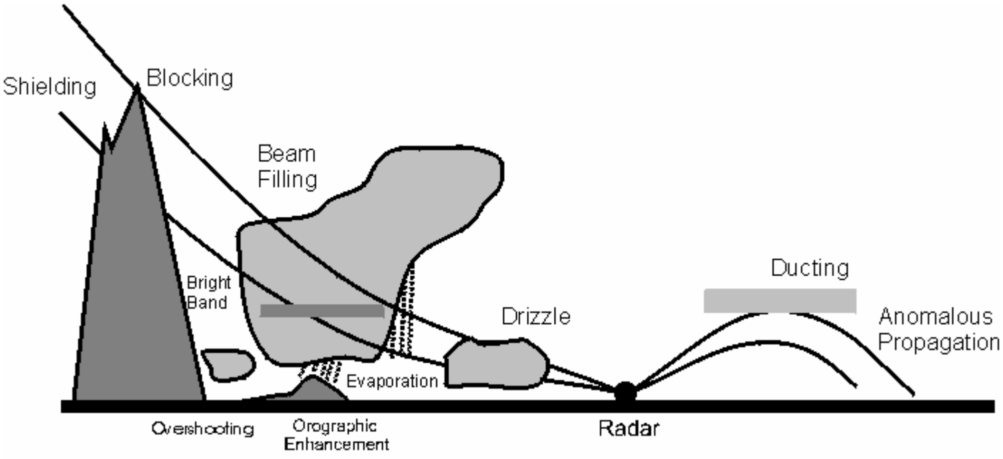


**Figure E.14: Depiction of radar artifacts. (Source: Wikipedia)**

SPAS is designed to overcome many of these short-comings by carefully using radar data for defining the spatial patterns and relative magnitudes of precipitation, but allowing measured precipitation values ("ground truth") at gauges to govern the magnitude. When absolutely necessary, the observed precipitation values at gauges are nudged up (or down) to force SPAS results to be consistent with observed gauge values. Nudging gauge precipitation values helps to promote better consistency between the gauge value and the grid-cell value, even though these two values sometimes should not be the same since they are sampling different area sizes. For reasons discussed in the "SPAS versus Gauge Precipitation" section, the gauge value and grid-cell value can vary. Plus, SPAS is designed to toss observed individual hourly values that are grossly inconsistent with radar data, hence driving a difference between the gauge and grid-cell. In general, when the gauge and grid-cell value differ by more than 15% and/or 0.50 inches, and the gauge data have been validated, then it is justified to artificially increase or decrease slightly the observed gauge value to "force" SPAS to derive a grid-cell value equal to the observed value. Sometimes simply shifting the gauge location to an adjacent grid-cell resolves the problems. Regardless, a large gauge versus grid-cell difference is a "red flag" and sometimes the result of an erroneous gauge value or a mis-located gauge, but in some cases the difference can only be resolved by altering the precipitation value.

Before results are finalized, a precipitation intensity check is conducted to ensure the spatial patterns and magnitudes of the maximum storm intensities at 1-, 6-, 12-, etc. hours are consistent with surrounding gauges and published reports. Any erroneous data are corrected and SPAS re-run. Considering all of the QA/QC checks in SPAS, it typically requires 5-15 basemap SPAS runs and, if radar data are available, another 5-15 radar-aided runs, to arrive at the final output.

## Test Cases

To check the accuracy of the DAD software, three test cases were evaluated.

### "Pyramidville" Storm

The first test was that of a theoretical storm with a pyramid shaped isohyetal pattern. This case was called the Pyramidville storm. It contained 361 hourly stations, each occupying a single grid-cell. The configuration of the Pyramidville storm (see Figure E.15) allowed for uncomplicated and accurate calculation of the analytical DA truth independent of the DAD software. The main motivation of this case was to verify that the DAD software was properly computing the area sizes and average depths.

1. Storm center: 39°N 104°W
2. Duration: 10-hours
3. Maximum grid-cell precipitation: 1.00”
4. Grid-cell resolution: 0.06 sq.-miles (361 total cells)
5. Total storm size: 23.11 sq-miles
6. Distribution of precipitation:

   Hour 1: Storm drops 0.10” at center (area 0.06 mi$^2$)

   Hour 2: Storm drops 0.10” over center grid-cell AND over one cell width around hour 1 center

   Hours 3-10:

   1. Storm drops 0.10” per hour at previously wet area, plus one cell width around previously wet area
   2. Area analyzed at every 0.10”
   3. Analysis resolution: 15-sec (~.25 mi$^2$)

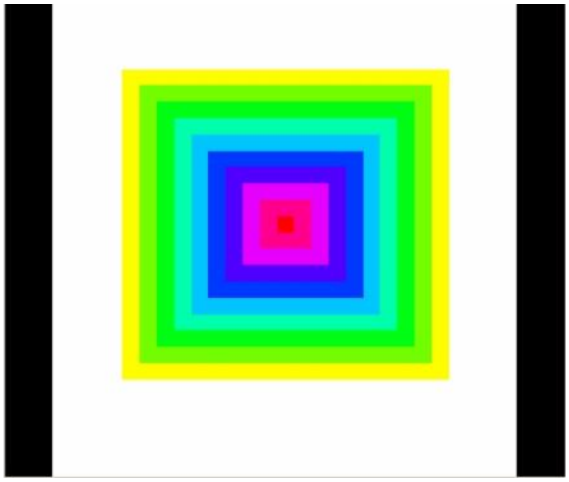


**Figure E.15: "Pyramidville” Total precipitation. Center = 1.00”, Outside edge = 0.10”**

The analytical truth was calculated independent of the DAD software, and then compared to the DAD output. The DAD software results were equal to the truth, thus demonstrating that the DA estimates were properly calculated (Figure E.16).

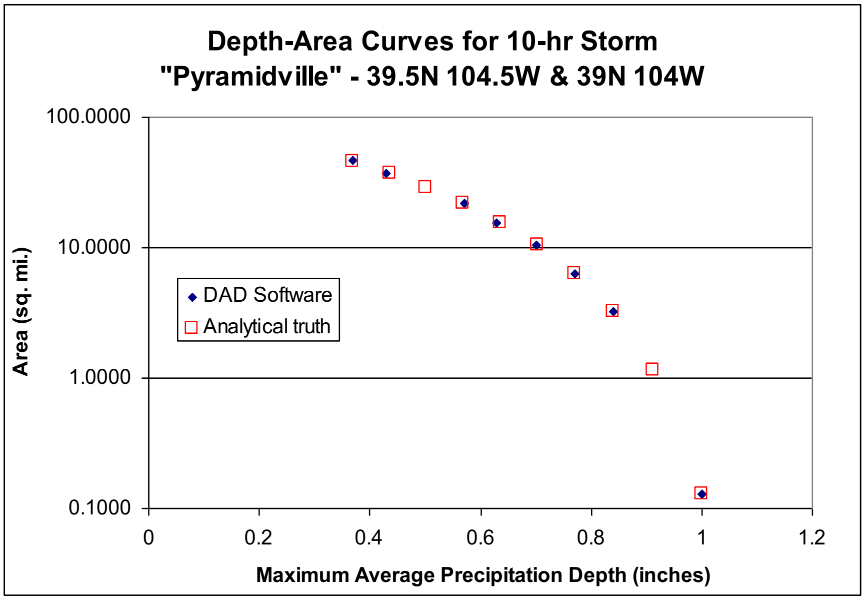


**Figure E.16: 10-hour DA results for "Pyramidville"; truth vs. output from DAD software**

The Pyramidville storm was then changed such that the mass curve and spatial interpolation methods would be stressed. Test cases included:

- Two-centers, each center with 361 hourly stations
- A single center with 36 hourly stations, 0 daily stations
- A single center with 3 hourly stations and 33 daily stations

As expected, results began shifting from the 'truth,' but minimally and within the expected uncertainty.

### Ritter, Iowa Storm, June 7, 1953

Ritter, Iowa was chosen as a test case for a number of reasons. The NWS had completed a storm analysis, with available DAD values for comparison. The storm occurred over relatively flat terrain, so orographics were not an issue. An extensive "bucket survey" provided a great number of additional observations from this event. Of the hundreds of additional reports, about 30 of the most accurate reports were included in the DAD analysis. The DAD software results are very similar to the NWS DAD values (Table E.2).

**Table E.2: The percent difference [(AWA-NWS)/NWS] between the AWA DA results and those published by the NWS for the 1953 Ritter, Iowa storm.**

**% Difference**

| | Duration (hours) | | | | |
|---|---|---|---|---|---|
| **Area (sq.mi.)** | | 6 | 12 | 24 | total |
| **10** | | -15% | -7% | 2% | 2% |
| **100** | | -7% | -6% | 1% | 1% |
| **200** | | 2% | 0% | 9% | 9% |
| **1000** | | -6% | -7% | 4% | 4% |
| **5000** | | -13% | -8% | 2% | 2% |
| **10000** | | -14% | -6% | 0% | 0% |

**Westfield, Massachusetts Storm, August 8, 1955**

Westfield, Massachusetts was also chosen as a test case for a number of reasons. It is a probable maximum precipitation (PMP) driver for the northeastern United States. Also, the Westfield storm was analyzed by the NWS and the DAD values are available for comparison. Although this case proved to be more challenging than any of the others, the final results are very similar to those published by the NWS (Table E.3).

**Table E.3: The percent difference [(AWA-NWS)/NWS] between the AWA DA results and those published by the NWS for the 1955 Westfield, Massachusetts storm**

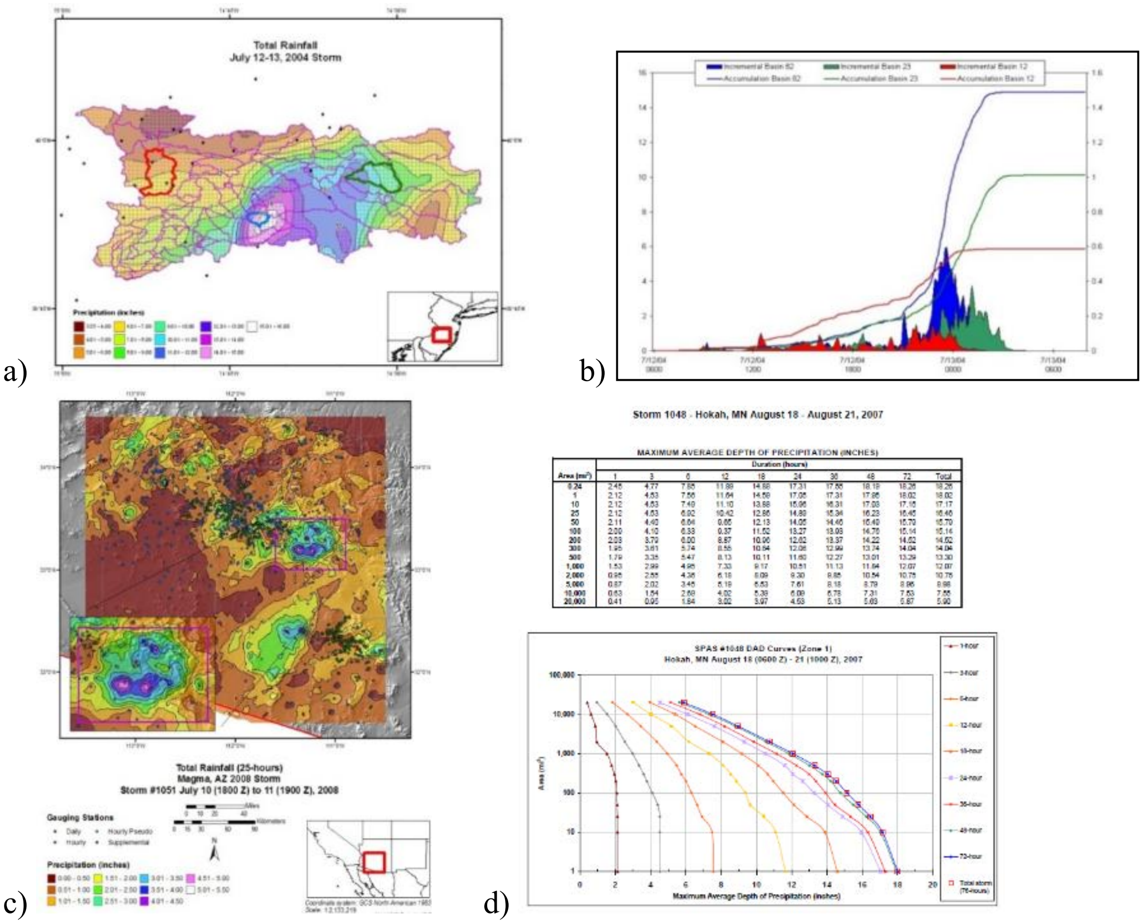
**% Difference**

| | **Duration (hours)** | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| **Area (sq. mi.)** | | 6 | 12 | 24 | 36 | 48 | 60 | total |
| | | | | | | | | |
| **10** | | 2% | 3% | 0% | 1% | -1% | 0% | 2% |
| **100** | | -5% | 2% | 4% | -2% | -6% | -4% | -3% |
| **200** | | -6% | 1% | 1% | -4% | -7% | -5% | -5% |
| **1000** | | -4% | -2% | 1% | -6% | -7% | -6% | -3% |
| **5000** | | 3% | 2% | -3% | -3% | -5% | -5% | 0% |
| **10000** | | 4% | 9% | -5% | -4% | -7% | -5% | 1% |
| **20000** | | 7% | 12% | -6% | -3% | -4% | -3% | 3% |

The primary components of SPAS are: storm search, data extraction, quality control (QC), conversion of daily precipitation data into estimated hourly data, hourly and total storm precipitation grids/maps and a complete storm-centered DAD analysis.

## Output

Armed with accurate, high-resolution precipitation grids, a variety of customized output can be created (see Figures E.17A-D). Among the most useful outputs are sub-hourly precipitation grids for input into hydrologic models. Sub-hourly (i.e. 5-minute) precipitation grids are created by applying the appropriate optimized hourly Z-R (scaled down to be applicable for instantaneous Z) to each of the individual 5-minute radar scans; 5-minutes is often the native scan rate of the radar in the US. Once the scaled Z-R is applied to each radar scan, the resulting precipitation is summed up. The proportion of each 5-minute precipitation to the total 1-hour radar-aided precipitation is calculated. Each 5-minute proportion (%) is then applied to the quality controlled, bias corrected 1-hour total precipitation (created above) to arrive at the final 5 minute precipitation for each scan. This technique ensures the sum of 5-minute precipitation equals that of the quality controlled, bias corrected 1-hour total precipitation derived initially. Depth-area-duration (DAD) tables/plots, shown in Figure E.17d, are computed using a highly-computational extension to SPAS. DADs provide an objective three dimensional (magnitude, area size, and duration) perspective of a storms' precipitation. SPAS DADs are computed using the procedures outlined by the NWS Technical Paper 1 (1946).



Storm 1048 - Hokah, MN August 18 - August 21, 2007

MAXIMUM AVERAGE DEPTH OF PRECIPITATION (INCHES)

| Area (mi²) | Duration (hours) | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | 1 | 3 | 6 | 12 | 18 | 24 | 36 | 48 | 72 | Total |
| 0.24 | 2.45 | 4.77 | 7.65 | 11.89 | 14.68 | 17.31 | 17.65 | 18.19 | 18.26 | 18.26 |
| 1 | 2.12 | 4.53 | 7.56 | 11.64 | 14.59 | 17.05 | 17.31 | 17.86 | 18.02 | 18.02 |
| 10 | 2.12 | 4.53 | 7.40 | 11.10 | 13.88 | 15.96 | 16.31 | 17.03 | 17.15 | 17.17 |
| 25 | 2.12 | 4.53 | 6.92 | 10.42 | 12.86 | 14.80 | 15.34 | 16.23 | 16.45 | 16.46 |
| 50 | 2.11 | 4.40 | 6.64 | 9.65 | 12.13 | 14.05 | 14.46 | 15.40 | 15.70 | 15.70 |
| 100 | 2.00 | 4.10 | 6.33 | 9.37 | 11.52 | 13.27 | 13.93 | 14.76 | 15.14 | 15.14 |
| 200 | 2.03 | 3.79 | 6.00 | 8.87 | 10.96 | 12.62 | 13.37 | 14.22 | 14.52 | 14.52 |
| 300 | 1.95 | 3.61 | 5.74 | 8.55 | 10.64 | 12.06 | 12.99 | 13.74 | 14.04 | 14.04 |
| 500 | 1.79 | 3.35 | 5.47 | 8.13 | 10.11 | 11.60 | 12.27 | 13.01 | 13.29 | 13.30 |
| 1,000 | 1.53 | 2.99 | 4.95 | 7.33 | 9.17 | 10.51 | 11.13 | 11.84 | 12.07 | 12.07 |
| 2,000 | 0.95 | 2.55 | 4.36 | 6.18 | 8.09 | 9.30 | 9.85 | 10.54 | 10.75 | 10.76 |
| 5,000 | 0.87 | 2.02 | 3.45 | 5.19 | 6.53 | 7.61 | 8.18 | 8.79 | 8.96 | 8.98 |
| 10,000 | 0.63 | 1.54 | 2.69 | 4.02 | 5.39 | 6.09 | 6.78 | 7.31 | 7.53 | 7.55 |
| 20,000 | 0.41 | 0.95 | 1.84 | 3.02 | 3.97 | 4.53 | 5.13 | 5.63 | 5.87 | 5.90 |

**Figure E.17: Various examples of SPAS output, including (a) total storm map and its associated (b) basin average precipitation time series, (c) total storm precipitation map, (d) depth-area-duration (DAD) table and plot**

# Summary

Grounded on years of scientific research with a demonstrated reliability in post-storm analyses, SPAS is a hydro-meteorological tool that provides accurate precipitation analyses for a variety of applications. SPAS has the ability to compute precise and accurate results by using sophisticated timing algorithms, basemaps, a variety of precipitation data and most importantly NEXRAD weather radar data (if available). The approach taken by SPAS relies on hourly, daily and supplemental precipitation gauge observations to provide quantification of the precipitation amounts while relying on basemaps and NEXRAD data (if available) to provide the spatial distribution of precipitation between precipitation gauge sites. By determining the most appropriate coefficients for the Z-R equation on an hourly basis, the approach anchors the precipitation amounts to accepted precipitation gauge data while using the NEXRAD data to distribute precipitation between precipitation gauges for each hour of the storm. Hourly Z-R coefficient computations address changes in the cloud microphysics and storm characteristics as the storm evolves. Areas suffering from limited or no radar coverage are estimated using the spatial patterns and magnitudes of the independently created basemap precipitation grids. Although largely automated, SPAS is flexible enough to allow hydro-meteorologists to make important adjustments and adapt to any storm situation.

# References

Baeck M.L., Smith J.A., 1998: "Precipitation Estimation by the WSR-88D for Heavy Precipitation Events", *Weather and Forecasting*: Vol. 13, No. 2, pp. 416–436.

Ciach, G.J., 2003: Local Random Errors in Tipping-Bucket Rain Gauge Measurements. *J. Atmos. Oceanic Technol.*, **20**, 752–759.

Corps of Engineers, U.S. Army, 1945-1973: Storm Rainfall in the United States, Depth-Area-Duration Data. Office of Chief of Engineers, Washington, D.C.

Corrigan, P., Fenn, D.D., Kluck, D.R., and J.L. Vogel, 1999: Probable Maximum Precipitation Estimates for California. *Hydrometeorological Report No. 59*, U.S. National Weather Service, National Oceanic and Atmospheric Administration, U.S. Department of Commerce, Silver Spring, MD, 392 pp.

Dickens, J., 2003: "On the Retrieval of Drop Size Distribution by Vertically Pointing Radar", American Meteorological Society 32nd Radar Meteorology Conference, Albuquerque, NM, October 2005.

Duchon, C.E., and G.R. Essenberg, 2001: Comparative Precipitation Observations from Pit and Above Ground Rain Gauges with and without Wind Shields, *Water Resources Research,* Vol. 37, N. 12, 3253-3263.

Faulkner, E., T. Hampton, R.M. Rudolph, and Tomlinson, E.M., 2004: Technological Updates for PMP and PMF – Can They Provide Value for Dam Safety Improvements? Association of State Dam Safety Officials Annual Conference, Phoenix, Arizona, September 26-30, 2004.

Guo, J. C. Y., Urbonas, B., and Stewart, K., 2001: Rain Catch under Wind and Vegetal Effects. ASCE, *Journal of Hydrologic Engineering*, Vol. 6, No. 1.

Hansen, E.M., Fenn, D.D., Schreiner, L.C., Stodt, R.W., and J.F., Miller, 1988: Probable Maximum Precipitation Estimates, United States between the Continental Divide and the 103$^{rd}$ Meridian, *Hydrometeorological Report Number 55A*, National weather Service, National Oceanic and Atmospheric Association, U.S. Dept of Commerce, Silver Spring, MD, 242 pp.

Hunter, R.D. and R.K. Meentemeyer, 2005: Climatologically Aided Mapping of Daily Precipitation and Temperature, *Journal of Applied Meteorology*, October 2005, Vol. 44, pp. 1501-1510.

Hunter, S.M., 1999: Determining WSR-88D Precipitation Algorithm Performance Using The Stage III Precipitation Processing System, Next Generation Weather Radar Program, WSR-88D Operational Support Facility, Norman, OK.

Lakshmanan, V. and M. Valente, 2004: Quality control of radar reflectivity data using satellite data and surface observations, 20th Int'l Conf. on Inter. Inf. Proc. Sys. (IIPS) for Meteor., Ocean., and Hydr., Amer. Meteor. Soc., Seattle, CD-ROM, 12.2.

Martner, B.E, and V. Dubovskiy, 2005: Z-R Relations from Raindrop Disdrometers: Sensitivity To Regression Methods And DSD Data Refinements, 32nd Radar Meteorology Conference, Albuquerque, NM, October, 2005

Tokay, A., P.G. Bashor, and V.L. McDowell, 2010: Comparison of Rain Gauge Measurements in the Mid-Atlantic Region. *J. Hydrometeor.*, **11**, 553-565.

Tomlinson, E.M., W.D. Kappel, T.W. Parzybok, B. Rappolt, 2006: Use of NEXRAD Weather Radar Data with the Storm Precipitation Analysis System (SPAS) to Provide High Spatial

Resolution Hourly Precipitation Analyses for Runoff Model Calibration and Validation, ASDSO Annual Conference, Boston, MA.

Tomlinson, E.M., and T.W. Parzybok, 2004: Storm Precipitation Analysis System (SPAS), proceedings of Association of Dam Safety Officials Annual Conference, Technical Session II, Phoenix, Arizona.

Tomlinson, E.M., R.A. Williams, and T.W. Parzybok, September 2003: Site-Specific Probable Maximum Precipitation (PMP) Study for the Great Sacandaga Lake / Stewarts Bridge Drainage Basin, Prepared for Reliant Energy Corporation, Liverpool, New York.

Tomlinson, E.M., R.A. Williams, and T.W. Parzybok, September 2003: Site-Specific Probable Maximum Precipitation (PMP) Study for the Cherry Creek Drainage Basin, Prepared for the Colorado Water Conservation Board, Denver, CO.

Tomlinson, E.M., Kappel W.D., Parzybok, T.W., Hultstrand, D., Muhlestein, G., and B. Rappolt, May 2008: Site-Specific Probable Maximum Precipitation (PMP) Study for the Wanahoo Drainage Basin, Prepared for Olsson Associates, Omaha, Nebraska.

Tomlinson, E.M., Kappel W.D., Parzybok, T.W., Hultstrand, D., Muhlestein, G., and B. Rappolt, June 2008: Site-Specific Probable Maximum Precipitation (PMP) Study for the Blenheim Gilboa Drainage Basin, Prepared for New York Power Authority, White Plains, NY.

Tomlinson, E.M., Kappel W.D., and T.W. Parzybok, February 2008: Site-Specific Probable Maximum Precipitation (PMP) Study for the Magma FRS Drainage Basin, Prepared for AMEC, Tucson, Arizona.

Tomlinson, E.M., Kappel W.D., Parzybok, T.W., Hultstrand, D., Muhlestein, G., and P. Sutter, December 2008: Statewide Probable Maximum Precipitation (PMP) Study for the state of Nebraska, Prepared for Nebraska Dam Safety, Omaha, Nebraska.

Tomlinson, E.M., Kappel, W.D., and Tye W. Parzybok, July 2009: Site-Specific Probable Maximum Precipitation (PMP) Study for the Scoggins Dam Drainage Basin, Oregon.

Tomlinson, E.M., Kappel, W.D., and Tye W. Parzybok, February 2009: Site-Specific Probable Maximum Precipitation (PMP) Study for the Tuxedo Lake Drainage Basin, New York.

Tomlinson, E.M., Kappel, W.D., and Tye W. Parzybok, February 2010: Site-Specific Probable Maximum Precipitation (PMP) Study for the Magma FRS Drainage Basin, Arizona.

Tomlinson, E.M., Kappel W.D., Parzybok, T.W., Hultstrand, D.M., Muhlestein, G.A., March 2011: Site-Specific Probable Maximum Precipitation Study for the Tarrant Regional Water District, Prepared for Tarrant Regional Water District, Fort Worth, Texas.

Tomlinson, E.M., Kappel, W.D., Hultstrand, D.M., Muhlestein, G.A., and T. W. Parzybok, November 2011: Site-Specific Probable Maximum Precipitation (PMP) Study for the Lewis River basin, Washington State.

Tomlinson, E.M., Kappel, W.D., Hultstrand, D.M., Muhlestein, G.A., and T. W. Parzybok, December 2011: Site-Specific Probable Maximum Precipitation (PMP) Study for the Brassua Dam basin, Maine.

U.S. Weather Bureau, 1946: Manual for Depth-Area-Duration analysis of storm precipitation. *Cooperative Studies Technical Paper No. 1*, U.S. Department of Commerce, Weather Bureau, Washington, D.C., 73pp

# Appendix F
## Storm Data (Separate Binding)

# Appendix G
## GIS PMP Tool Documentation

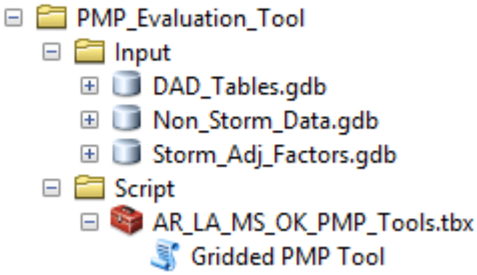
# 1. PMP Tool Description and Usage

The PMP Evaluation Tool employed in this study uses a Python-based script designed to run within the ArcGIS environment. ESRI's ArcGIS Desktop Basic software, version 1.4 or later, is required to run the tool, and it is recommended that the user have a basic familiarity with the operation of this software. The tool provides gridded PMP values at a spatial resolution of 90 arc-seconds (equivalent to .025 x .025 decimal degrees) for a user-designated drainage basin or area at user-specified durations, in addition to basin average PMP depths and temporally distributed accumulations.

## 1.1 File Structure

The tool, source script, and the storm databases are stored within the 'PMP_Evaluation_Tool' project folder. The file and directory structure within the 'PMP_Evaluation_Tool' folder should be maintained as it is provided, as the script will locate various data based on its relative location within the project folder. If the subfolders or geodatabases within are relocated or renamed, then the script must be updated to account for these changes.

The file structure consists of only two subfolders: Input and Script. The 'Input' folder contains all input GIS files (Figure 1.1). There are three ArcGIS file geodatabase containers within the 'Input' folder: DAD_Tables.gdb, Storm_Adj_Factors.gdb, and Non_Storm_Data.gdb. The DAD_Tables.gdb contains the DAD tables (in file geodatabase table format) for each of the SPAS-analyzed storm DAD zones included in the storm database. The Storm_Adj_Factors.gdb contains a feature class for each storm center and stores the adjustment factors for each grid point as a separate feature. These feature classes are organized into feature datasets, according to storm type (General, Local, and Tropical). The storm adjustment factor feature classes share their name with their DAD Table counterpart. The naming convention is SPAS_XXXX_Y, where XXXX is the SPAS storm ID number and Y is the DAD zone number. In the case of a hybrid storm (i.e., a storm that is run as both a general and local storm type), there will be a suffix "_gen" or "_loc" to differentiate the storm type specific to the adjustment factors in the feature class. Finally, the Non_Storm_Data.gdb contains spatial data not directly relating to the input rainfall depth or adjustment factors such as the grid network vector files. The geodatabase also contains the temporal distribution pattern tables and a table and a feature class of the storm center locations.



```
⊟ PMP_Evaluation_Tool
  ⊟ Input
    ⊞ DAD_Tables.gdb
    ⊞ Non_Storm_Data.gdb
    ⊞ Storm_Adj_Factors.gdb
  ⊟ Script
    ⊟ AR_LA_MS_OK_PMP_Tools.tbx
        Gridded PMP Tool
```

**Figure 1.1 - PMP tool file structure.**

The 'Script' folder contains an ArcToolbox called PMP_Tools.tbx. The toolbox contains a script tool called 'Gridded PMP Tool' that is used to calculate basin PMP. ArcGIS Desktop should be used for viewing the GIS tool file structure and interacting with the input and output geospatial data and metadata. A typical operating system’s file browser does not allow access to the geodatabase containers and cannot be used to directly run the tool.

## 1.2 Usage

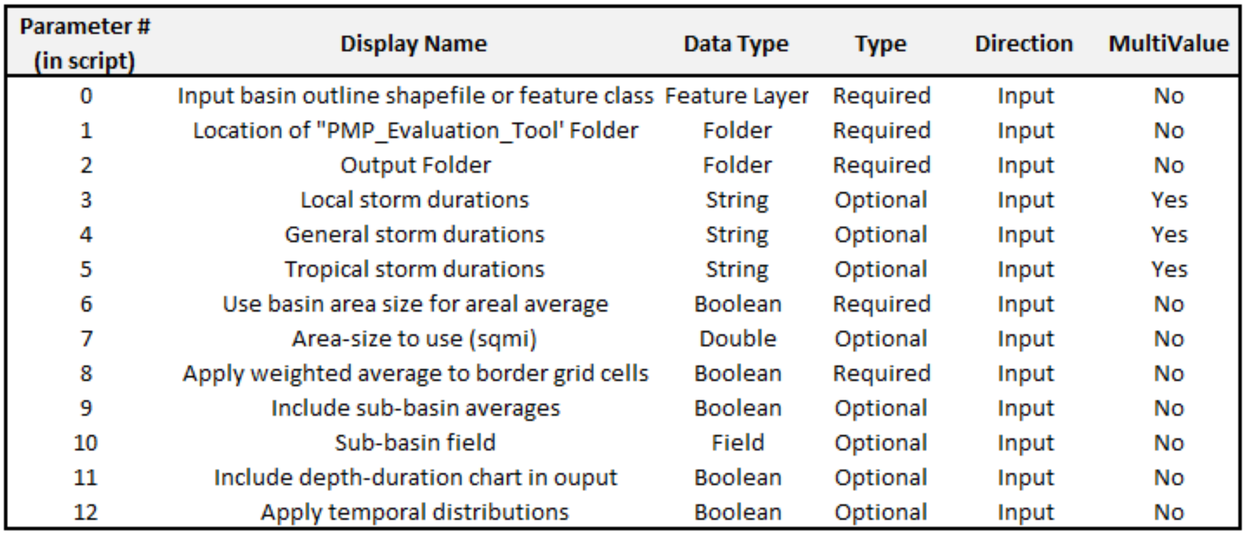
The ‘Gridded PMP Tool’ tool stored within the PMP_Tools.tbx. ArcToolbox opens and runs the script within the ArcGIS environment and can be run from ArcCatalog an ArcMap map session. In addition to running as a standalone tool, the tool can be incorporated into Model Builder or be called as a sub-function of another script.

To run the tool, the user navigates to the PMP_Tools.tbx toolbox, expands it, and opens the PMP tool. The dialogue window opens and the user populates input parameters (see Figure 1.2) and clicks the ‘OK’ button. The tool will run in the foreground and display text output in the Messages window. Processing time can vary greatly depending on AOI size, the number of durations selected, and computer hardware. Most basins generally take 5 to 10 minutes to analyze all three storm types. The tool produces PMP output described in Section 1.4.

## 1.3 Input Parameters

The tool requires twelve parameters as input to define the area and durations to be analyzed ( Table **1.1**).

**Table 1.1 - Parameters for the PMP calculation tool.**

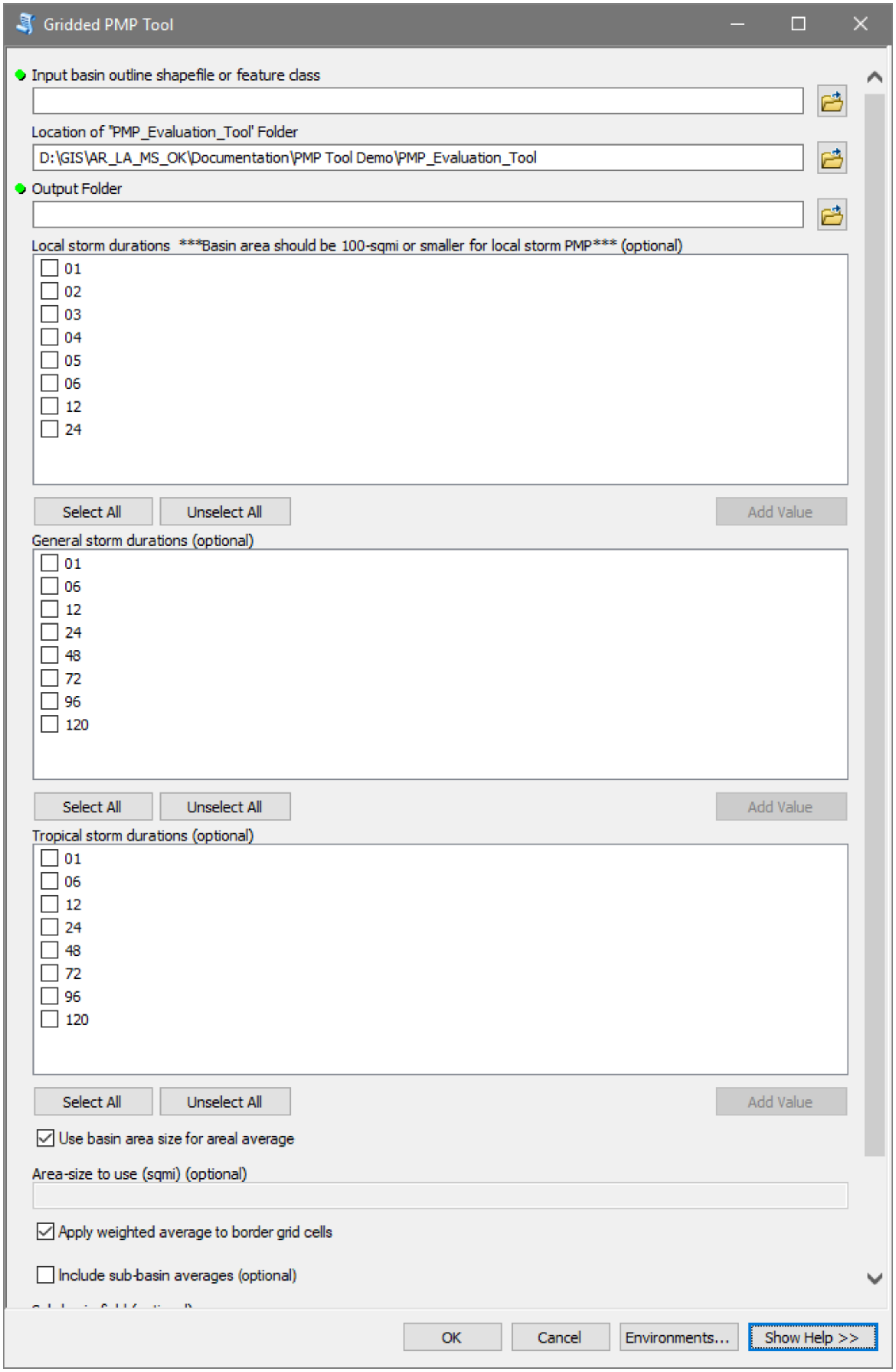


| Parameter # (in script) | Display Name | Data Type | Type | Direction | MultiValue |
|---|---|---|---|---|---|
| 0 | Input basin outline shapefile or feature class | Feature Layer | Required | Input | No |
| 1 | Location of "PMP_Evaluation_Tool' Folder | Folder | Required | Input | No |
| 2 | Output Folder | Folder | Required | Input | No |
| 3 | Local storm durations | String | Optional | Input | Yes |
| 4 | General storm durations | String | Optional | Input | Yes |
| 5 | Tropical storm durations | String | Optional | Input | Yes |
| 6 | Use basin area size for areal average | Boolean | Required | Input | No |
| 7 | Area-size to use (sqmi) | Double | Optional | Input | No |
| 8 | Apply weighted average to border grid cells | Boolean | Required | Input | No |
| 9 | Include sub-basin averages | Boolean | Optional | Input | No |
| 10 | Sub-basin field | Field | Optional | Input | No |
| 11 | Include depth-duration chart in ouput | Boolean | Optional | Input | No |
| 12 | Apply temporal distributions | Boolean | Optional | Input | No |

Figure 1.2 shows the tool dialogue window with each of the input parameters. The first parameter required by the tool dialogue is a feature layer, such as a basin shapefile or feature class, designed to outline the area of interest (AOI) for the PMP analysis. If the AOI dataset does not have a surface projection, the tool will apply the Albers Equal Area projection for the purpose of calculating the AOI area size. If the feature layer has multiple features (or polygons), the tool will use the combined area as the analysis region. Only the selected polygons will be used if the tool is run from the ArcMap environment with selected features highlighted. If the AOI shapefile extends beyond the project analysis domain PMP will only be calculated for grid

cells inside the project domain. The AOI shapefile or feature class should not have any spaces or symbol characters in the filename.

The second parameter requires the path of the ‘PMP_Evaluation_Tool’ folder. The default location of the folder is set within the tool parameters, but it can be changed if the user wishes to link the tool to another set of input datasets. The ‘PMP_Evaluation_Tool’ project folder should be stored locally at a location that can be accessed (both read/write) by ArcGIS desktop. The user will need to set the ‘Output Folder’ path which provides the tool with the location to create the output PMP files. The user must have read/write privileges for this folder location. The user then selects the durations to be run for each storm type. The next parameter allows the user may override the default to use the input basin feature area size for areal-average PMP calculations and enter a custom area (in square miles). A manually entered area-size will override the basin area-size in the PMP calculations. Next, the user has the option to have the tool perform a weighted analysis on the grid cells underlying the AOI boundary. If this option is checked each boundary grid cell depth will be weighted by the portion of the cell’s area inside the basin for the purposed of the basin area PMP table calculations. Finally, there is an option to include sub-basin averages. This will calculate an average PMP depth for each feature in the input basin feature class from the overall basin PMP. The average sub-basin depths will be based on the area-size of the overall basin. If the ‘weighted’ option was selected above it will also be applied to the sub-basin averages. If the AOI has multiple features (or sub-basins) the ‘Include sub-basin averages’ option can be selected to calculate the basin average PMP over each sub-basin. The user must select a field within the AOI to be used to identify each sub-basin. The field can be of numeric or text data type, but must have a unique ID for each polygon. The user can also choose to include a depth-duration chart .png image in the output folder for each storm type. Finally, the user can select the option to apply the appropriate temporal distribution patterns to the basin average PMP for each storm type. If this option is selected, the above options to select specific durations will no longer be available as the tool will need to run all the durations to properly apply temporal distributions.



● Input basin outline shapefile or feature class

Location of 'PMP_Evaluation_Tool' Folder

D:\GIS\AR_LA_MS_OK\Documentation\PMP Tool Demo\PMP_Evaluation_Tool

● Output Folder

Local storm durations ***Basin area should be 100-sqmi or smaller for local storm PMP*** (optional)

□ 01
□ 02
□ 03
□ 04
□ 05
□ 06
□ 12
□ 24

General storm durations (optional)

□ 01
□ 06
□ 12
□ 24
□ 48
□ 72
□ 96
□ 120

Tropical storm durations (optional)

□ 01
□ 06
□ 12
□ 24
□ 48
□ 72
□ 96
□ 120

☑ Use basin area size for areal average

Area-size to use (sqmi) (optional)

☑ Apply weighted average to border grid cells

□ Include sub-basin averages (optional)

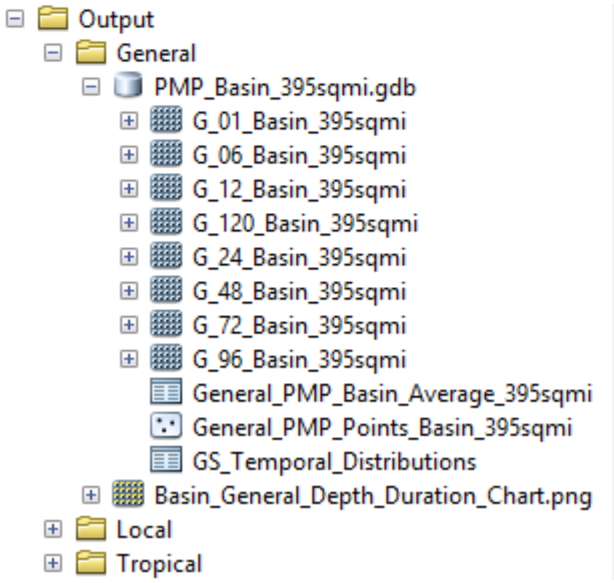
**Figure 1.2: The PMP Evaluation Tool input dialogue window**

The Validation tab of the tool properties contains some custom scripting to handle the input parameter formatting.

### 1.4 Tool Output

Once the tool has been run, the output file geodatabases will be populated with the model results. The GIS files can then be brought into an ArcMap, or other compatible GIS environments, for mapping and analysis. The tool is set to have overwrite capabilities; if output data exists, it will be overwritten the next time the tool is run, if the same output folder and same parameters are used.

A separate output folder is created for each storm type and the output is organized within file geodatabases and named according to the input basin feature name and analyzed PMP area. Each output file geodatabase contains a feature class which stores each grid point centroid within the basin as a separate feature. Each feature has a field for the grid ID, latitude, longitude, analysis zone, elevation, PMP (for each duration), and the contributing storm ID. The PMP raster files are also stored within the file geodatabase. The naming convention for the raster files is the storm type and duration (L for local/MCS, G for general, and T for tropical), followed by the input basin feature name, and ending with the basin area (in square miles). An example of the output file structure is shown in Figure 1.3.



```
⊟ Output
    ⊟ General
        ⊟ PMP_Basin_395sqmi.gdb
            ⊞ G_01_Basin_395sqmi
            ⊞ G_06_Basin_395sqmi
            ⊞ G_12_Basin_395sqmi
            ⊞ G_120_Basin_395sqmi
            ⊞ G_24_Basin_395sqmi
            ⊞ G_48_Basin_395sqmi
            ⊞ G_72_Basin_395sqmi
            ⊞ G_96_Basin_395sqmi
              General_PMP_Basin_Average_395sqmi
              General_PMP_Points_Basin_395sqmi
              GS_Temporal_Distributions
        ⊞ Basin_General_Depth_Duration_Chart.png
    ⊞ Local
    ⊞ Tropical
```

**Figure 1.3: Example of the PMP tool output file structure**

### 1.5 Python Script

Due to the large number of storm datasets and grid points within the project domain, a scripted process is well suited for comparing each value efficiently and accurately for a given area of interest and make the necessary calculations. ArcGIS has integrated the Python scripting language to allow for the custom development of geoprocessing operations and toolsets. Python can be used to access the geoprocessing, data management, and looping functionality needed to process the PMP calculations for a basin.  The gridded PMP analysis script has been added to an ArcToolbox and can be run as a tool within the ArcGIS environment. The script has been imported and stored internally within the Gridded PMP Tool and all the parameters for the tool have been set. The script can be accessed by exporting it from the tool to a '.py' file. The Python code can be opened and edited within any text editor. A hardcopy version of the code is given in Appendix H.

The python script uses the arcpy, arcpy.analysis, arcpy.management, arcpy.conversion, numpy, pandas, and matplotlib.pyplot modules. Python and these modules are included within the ArcGIS for Desktop package. The script is designed to run as efficiently as possible with a minimal amount of code and complexity. To achieve this, the script is organized into functions that are called as needed. The primary PMP analysis calculations are calculated within the pmpAnalysis() function which is called separately for each PMP storm type analyzed. Within the broader pmpAnalysis() function, several smaller functions are called to perform various tasks:

| | |
|---|---|
| createPMPfc() | Creates the PMP_Points feature class to store vector (point) results |
| getAOIarea() | Calculates the area of the input basin |
| dadLookup() | Gets the DAD value for the current storm based on basin area or AOI defined by the user |
| updatePMP() | Records the largest adjusted rainfall value (PMP) |
| outputPMP() | Produces output PMP GIS files and tables |
| basinAve() | Calculates the basin average PMP |
| basinZone() | Returns the transposition zone and side of Continental Divide of AOI for the temporal distribution application |
| temporalDistLS() | Local storm temporal distribution application |
| temporalDistGS() | General storm temporal distribution application |
| temporalDistTS() | Tropical storm temporal distribution application |
| temporalCritStaced() | Critically stacked temporal distribution application |
| checkTemporal() | Produces a table evaluating exceedance errors in temporally distributed depths at intermediate durations |

There is extensive documentation within the code in the form of '# comments'. These comments provide guidance toward its functionality and describe the code.

While the script performs many actions, its primary purpose is to iterate through both the storm list and the grid points within the drainage basin area of interest (AOI), comparing each, and creating output based on the maximum values. To accomplish this, several layers of nested iterative "for" loops are used.

The following high-level algorithm broadly describes the script process:

- Calculate Basin Area (in mi$^2$)
- For each Storm Type (general, tropical, and local)
    - For each duration
        - For each storm in database
            - Lookup storm's depth-area-duration (DAD) value for AOI size
            - For each grid point in basin
                - Calculate total adjusted rainfall (TAR) by multiplying DAD value by total adjustment factor for the grid point

        - If TAR > PMP, the TAR becomes the new PMP value for that grid point
    - Create PMP point feature class for the storm type
    - Create PMP raster GRID files for each duration
    - Create basin/sub-basin average tables
    - Create a depth-duration chart
    - Apply temporal distributions
    - Check temporal output for exceedance errors

### 1.6 Known Issues and Troubleshooting

The following guidelines may prevent issues with running the GIS tool.

- Ensure you version of ArcGIS Desktop is version 1.4 (or later) and maintenance is current.
- Ensure all file and path names do not have spaces or non-alphanumeric symbols (e.g. #, $, %). Underscores are acceptable and a good alternative to using spaces.
- Close any other applications or instances of ArcMap that may interfere with the current session, files, or file paths that will be used by the tool.
- Ensure that the tool folder, output location, and AOI files are all stored locally and not over a network location.

If the points above have been verified and issues persist, the user may try the following actions to address the issue:

- Close out ArcMap session and all open ArcGIS applications and restart session.
- Restart computer. This may be required to completely clear any locks on files or memory.
- Run the Repair Geometry tool on the AOI shapefile or feature class to correct any geometry issues within the file.
- Rename AOI file. Change tool and/or output folder paths.

If issues persist it may be necessary to contact ESRI support or perform a clean ArcGIS installation or upgrade

# Appendix H
## GIS Tool Python Script

```
''' ---------------------------------------------------------------------------
Name: Gridded PMP Tool Python Script

Script Version: 1.9

Python Version: 2.7

ArcGIS Version: ArcGIS Desktop 10.5.1

Author: Applied Weather Associates

Usage:  The tool is designed to be executed within an ArcMap environment with an open MXD
session.

Description:
    This tool calculates PMP depths for a given drainage basin for the
specified durations.  PMP point values are calculated (in inches) for each
grid point (spaced at 90 arc-second intervals) over the input polygon within
the project domain. The points are converted to gridded PMP datasets for
each duration.

---------------------------------------------------------------------------'''

###########################################################################
## import Python modules

import sys
import arcpy
import os
import traceback
from arcpy import env
import arcpy.analysis as an
import arcpy.management as dm
import arcpy.conversion as con
import numpy as np
import pandas as pd
from pandas import ExcelFile
import matplotlib.pyplot as plt

env.overwriteOutput = True                                                # Set overwrite option
env.addOutputsToMap = False

###########################################################################
## get input parameters

basin = arcpy.GetParameter(0)                                                   # get AOI Basin
Shapefile
home = arcpy.GetParameterAsText(1)                                              # get location of
'PMP' Project Folder
outLocation = arcpy.GetParameterAsText(2)
locDurations = arcpy.GetParameter(3)                                            # get local storm
durations (string)
genDurations = arcpy.GetParameter(4)                                            # get general
storm durations (string)
tropDurations = arcpy.GetParameter(5)                                           # get tropical
storm durations (string)
weightedAve = arcpy.GetParameter(8)                                        # get option to apply
weighted average (boolean)
#outputTable = arcpy.GetParameter(9)                                       # get file path for
basin average summary table
includeSubbasin = arcpy.GetParameter(9)                                    # get option add
subbasin averages (boolean)
subbasinIDfield = arcpy.GetParameterAsText(10)                                     # Subbasin ID
field from AOI Basin Shapefile
ddChart = arcpy.GetParameter(11)                                           # get option to create
depth-duration chart(boolean)
runTemporal = arcpy.GetParameter(12)                                       # get option to run
temporal distributions (boolean)
```

```
dadGDB = home + "\\Input\\DAD_Tables.gdb"                                       # location of DAD
tables
adjFactGDB = home + "\\Input\\Storm_Adj_Factors.gdb"                            # location of
feature datasets containing total adjustment factors
arcpy.AddMessage("\nDAD Tables geodatabase path:  " + dadGDB)
arcpy.AddMessage("Storm Adjustment Factor geodatabase path:  " + adjFactGDB)

#mxd = arcpy.mapping.MapDocument("CURRENT")
#df = arcpy.mapping.ListDataFrames(mxd)[0]
basAveTables = []                                                               # global list of
Basin Average Summary tables

def pmpAnalysis(aoiBasin, stormType, durList):

    ###########################################################################
    ## Create PMP Point Feature Class from points within AOI basin and add fields
    def createPMPfc():

        arcpy.AddMessage("\nCreating feature class: 'PMP_Points' in Scratch.gdb...")
        dm.MakeFeatureLayer(home + "\\Input\\Non_Storm_Data.gdb\\Vector_Grid", "vgLayer")
# make a feature layer of vector grid cells
        dm.SelectLayerByLocation("vgLayer", "INTERSECT", aoiBasin)
# select the vector grid cells that intersect the aoiBasin polygon
        dm.MakeFeatureLayer(home + "\\Input\\Non_Storm_Data.gdb\\Grid_Points", "gpLayer")
# make a feature layer of grid points
        dm.SelectLayerByLocation("gpLayer", "HAVE_THEIR_CENTER_IN", "vgLayer")
# select the grid points within the vector grid selection
        con.FeatureClassToFeatureClass("gpLayer", env.scratchGDB, "PMP_Points")
# save feature layer as "PMP_Points" feature class
        arcpy.AddMessage("(" + str(dm.GetCount("gpLayer")) + " grid points will be analyzed)\n")

        # Add PMP Fields
        for dur in durList:
            arcpy.AddMessage("\t...adding field: PMP_" + str(dur))
            dm.AddField(env.scratchGDB + "\\PMP_Points", "PMP_" + dur, "DOUBLE")

        # Add STORM Fields (this string values identifies the driving storm by SPAS ID number)
        for dur in durList:
            arcpy.AddMessage("\t...adding field: STORM_" + str(dur))
            dm.AddField(env.scratchGDB + "\\PMP_Points", "STORM_" + dur, "TEXT", "", "", 16,
"Storm ID " + dur + "-hour")

        # Add STNAME Fields (this string values identifies the driving storm by SPAS ID number)
        for dur in durList:
            arcpy.AddMessage("\t...adding field: STNAME_" + str(dur))
            dm.AddField(env.scratchGDB + "\\PMP_Points", "STNAME_" + dur, "TEXT", "", "", 50,
"Storm Name " + dur + "-hour")

        return

    ###########################################################################
    ##  Define getAOIarea() function:
    ##  getAOIarea() calculates the area of AOI (basin outline) input shapefile/
    ##  featureclass.  The basin outline shapefile must be projected.  The area
    ##  is sqaure miles, converted from the basin layers projected units (feet
    ##  or meters).  The aoiBasin feature class should only have a single feature
    ##  (the basin outline).  If there are multiple features, the area will be stored
    ##  for the final feature only.

    def getAOIarea():
        sr = arcpy.Describe(aoiBasin).SpatialReference                                          #
Determine aoiBasin spatial reference system
        srname = sr.name
        srtype = sr.type
        srunitname = sr.linearUnitName                                                          #
Units
        arcpy.AddMessage("\nAOI basin spatial reference:  " + srname + "\nUnit type: " +
srunitname + "\nSpatial reference type: " + srtype)

        aoiArea = 0.0
        rows = arcpy.SearchCursor(aoiBasin)
        for row in rows:
            feat = row.getValue("Shape")
            aoiArea += feat.area
        if srtype == 'Geographic':                                  # Must have a surface
projection.  If one doesn't exist it projects a temporary file and uses that.
            arcpy.AddMessage("\n***The basin shapefile's spatial reference 'Geographic' is not
supported.  Projecting temporary shapefile for AOI.***")
            arcpy.Project_management(aoiBasin,env.scratchGDB + "\\TempBasin",102039)     #
Projects AOI Basin (102039 = USA_Contiguous_Albers_Equal_Area_Conic_USGS_version)
            TempBasin = env.scratchGDB + "\\TempBasin"                                    # Path
to temporary basin created in scratch geodatabase
            sr = arcpy.Describe(TempBasin).SpatialReference                        # Determine
Spatial Reference of temporary basin
            aoiArea = 0.0
            rows = arcpy.SearchCursor(TempBasin)                                          #
Assign area size in square meters
            for row in rows:
                feat = row.getValue("Shape")
                aoiArea += feat.area
            aoiArea = aoiArea * 0.000000386102                                           #
Converts square meters to square miles
        elif srtype == 'Projected':
            if srunitname == "Meter":
                aoiArea = aoiArea * 0.000000386102                                              #
Converts square meters to square miles
            elif srunitname == "Foot" or "Foot_US":
                aoiArea = aoiArea * 0.00000003587                                               #
Converts square feet to square miles
            else:
                arcpy.AddMessage("\nThe basin shapefile's unit type '" + srunitname + "' is not
supported.")
                sys.exit("Invalid linear units")                                                #
Units must be meters or feet

        aoiArea = round(aoiArea, 3)
        arcpy.AddMessage("\nArea of interest: " + str(aoiArea) + " square miles.")

        if arcpy.GetParameter(6) == False:
            aoiArea = arcpy.GetParameter(7)                                                     #
Enable a constant area size
        aoiArea = round(aoiArea, 1)
        arcpy.AddMessage("\n***Area used for PMP analysis: " + str(aoiArea) + " sqmi***")
        return aoiArea

    ###########################################################################
    ##  Define dadLookup() function:
    ##  The dadLookup() function determines the DAD value for the current storm
    ##  and duration according to the basin area size.  The DAD depth is interpolated
    ##  linearly between the two nearest areal values within the DAD table.
    def dadLookup(stormLayer, duration, area):                  # dadLookup() accepts the current
storm layer name (string), the current duration (string), and AOI area size (float)
        #arcpy.AddMessage("\t\tfunction dadLookup() called.")
        durField = "H_" + duration                              # defines the name of the
duration field (eg., "H_06" for 6-hour)
        dadTable = dadGDB + "\\" + stormLayer
        rows = arcpy.SearchCursor(dadTable)

        try:
            row = rows.next()                                       # Sets DAD area x1 to the
value in the first row of the DAD table.
            x1 = row.AREASQMI
            y1 = row.getValue(durField)
            xFlag = "FALSE"                                         # xFlag will remain false for
basins that are larger than the largest DAD area.
        except RuntimeError:                                        # return if duration does not
exist in DAD table
            return

        row = rows.next()
        i = 0
        while row:                                                  # iterates through the DAD 
table - assiging the bounding values directly above and below the basin area size
            i += 1
            if row.AREASQMI < area:
                x1 = row.AREASQMI
                y1 = row.getValue(durField)
            else:
                xFlag = "TRUE"                                      # xFlag is switched to "TRUE" 
indicating area is within DAD range
                x2 = row.AREASQMI
                y2 = row.getValue(durField)
                break

            row = rows.next()
        del row, rows, i

        if xFlag == "FALSE":
            x2 = area                                           # If x2 is equal to the basin 
area, this means that the largest DAD area is smaller than the basin and the resulting DAD value 
must be extrapolated.
            arcpy.AddMessage("\t\tThe basin area size: " + str(area) + " sqmi is greater than the 
largest DAD area: " + str(x1) + " sqmi.\n\t\tDAD value is estimated by extrapolation.")
            y = x1 / x2 * y1                                    # y (the DAD depth) is estimated 
by extrapolating the DAD area to the basin area size.
            return y                                            # The extrapolated DAD depth (in 
inches) is returned.

        # arcpy.AddMessage("\nArea = " + str(area) + "\nx1 = " + str(x1) + "\nx2 = " + str(x2) + 
"\ny1 = " + str(y1) + "\ny2 = " + str(y2))

        x = area                                                # If the basin area size is 
within the DAD table area range, the DAD depth is interpolated
        deltax = x2 - x1                                        # to determine the DAD value (y) 
at area (x) based on next lower (x1) and next higher (x2) areas.
        deltay = y2 - y1
        diffx = x - x1

        y = y1 + diffx * deltay / deltax

        if x < x1:
            arcpy.AddMessage("\t\tThe basin area size: " + str(area) + " sqmi is less than the 
smallest DAD table area: " + str(x1) + " sqmi.\n\t\tDAD value is estimated by extrapolation.")

        return y                                                # The interpolated DAD depth (in 
inches) is returned.

    ###########################################################################
    ##  Define updatePMP() function:
    ##  This function updates the 'PMP_XX_' and 'STORM_XX' fields of the PMP_Points
    ##  feature class with the largest value from all analyzed storms stored in the
    ##  pmpValues list.
    def updatePMP(pmpValues, stormID, duration):
# Accepts four arguments: pmpValues - largest adjusted rainfall for current duration (float 
list); stormID - driver storm ID for each PMP value (text list); and duration (string)
##        pmpfield = "PMP_" + duration
##        stormfield = "STORM_" + duration
##        gridRows = arcpy.UpdateCursor(env.scratchGDB + "\\PMP_Points")
# iterates through PMP_Points rows
##        i = 0
##        for row in gridRows:
##            row.setValue(pmpfield, pmpValues[i])
# Sets the PMP field value equal to the Max Adj. Rainfall value (if larger than existing value).
##            row.setValue(stormfield, stormID[i])
# Sets the storm ID field to indicate the driving storm event
##            gridRows.updateRow(row)
##            i += 1
##        del row, gridRows, pmpfield, stormfield
##        arcpy.AddMessage("\n\t" + duration + "-hour PMP values update complete. \n")
##        return

        pmpfield = "PMP_" + duration
        stormfield = "STORM_" + duration
        stormTextField = "STNAME_" + duration

        gridRows = env.scratchGDB + "\\PMP_Points"
# iterates through PMP_Points rows
        i = 0
        with arcpy.da.UpdateCursor(gridRows, (pmpfield, stormfield, stormTextField)) as cursor:
            for row in cursor:
                row[0] = pmpValues[i]
# Sets the PMP field value equal to the Max Adj. Rainfall value (if larger than existing value).
                row[1] = stormID[i]
# Sets the storm ID field to indicate the driving storm event
                if not stormID[i] == "STORM":
# Conditional loop unless no controlling storm exists (ie, id of "STORM") to update table with
storm text
                    if stormID[i].endswith("_loc") or stormID[i].endswith("_gen") or
stormID[i].endswith("_tro"):   # remove hybrid storm suffixes
                        stormNum = stormID[i][:-4]
                    else:
                        stormNum = stormID[i]
                    whereClause = '"SPAS_ID"' + " = '" + stormNum + "'"
                    stormList = home + "\\Input\\Non_Storm_Data.gdb\\Storm_List"
                    sourceNameField = "STORM_TXT"
                    stormText = arcpy.da.SearchCursor(stormList, (sourceNameField,),
whereClause).next()[0]
                    row[2] = stormText
                    del stormNum, sourceNameField, stormList, stormText
                cursor.updateRow(row)
                i += 1
        del row, gridRows, pmpfield, stormfield, stormTextField, i
        arcpy.AddMessage("\n\t" + duration + "-hour PMP values update complete. \n")
        return

    ###########################################################################
    ##  The outputPMP() function produces raster GRID files for each of the PMP durations.
    ##  Aslo, a space-delimited PMP_Distribition.txt file is created in the 'Text_Output' folder.
    def outputPMP(type, area, outPath):
        desc = arcpy.Describe(basin)
        basinName = desc.baseName
        pmpPoints = env.scratchGDB + "\\PMP_Points"                             # Location of
'PMP_Points' feature class which will provide data for output

        outType = type[:1]
        outArea = str(int(round(area,0))) + "sqmi"
        outGDB = "PMP_"+ basinName + "_" + outArea +".gdb"
        if not arcpy.Exists(outPath + "\\" + outGDB):                           # Check to see if
PMP_XXXXX.gdb already exists
            arcpy.AddMessage("\nCreating output geodatabase '" + outGDB + "'")
            dm.CreateFileGDB(outPath, outGDB)
        arcpy.AddMessage("\nCopying PMP_Points feature class to " + outGDB + "...")
        con.FeatureClassToFeatureClass(pmpPoints, outPath + "\\" + outGDB, type + "_PMP_Points_"
+ basinName + "_" + outArea)
        pointFC = outPath + "\\" + outGDB + "\\" + type + "_PMP_Points_" + basinName + "_" +
outArea
        # addLayerMXD(pointFC)  # calls addLayerMDX function to add output to ArcMap session

        arcpy.AddMessage("\nBeginning PMP Raster Creation...")

        for dur in durList:                                                     # This code
creates a raster GRID from the current PMP point layer
            durField = "PMP_" + dur
            outLoc = outPath + outGDB +"\\" + outType + "_" + dur + "_" + basinName + "_" +
outArea
            arcpy.AddMessage("\n\tInput Path: " + pmpPoints)
            arcpy.AddMessage("\tOutput raster path: " + outLoc)
            arcpy.AddMessage("\tField name: " + durField)
            con.FeatureToRaster(pmpPoints, durField, outLoc, "0.025")
            arcpy.AddMessage("\tOutput raster created...")
        del durField, outLoc, dur

        arcpy.AddMessage("\nPMP Raster Creation complete.")
        if includeSubbasin:                                                 # Begin subbasin
average calculations
            subbasinID = []
            with arcpy.da.SearchCursor(basin, subbasinIDfield) as cursor:   # Create list of
subbasin ID names
                for row in cursor:
                    subbasinID.append(row[0])

            subIDtype = arcpy.ListFields(basin, subbasinIDfield)[0].type    # Define the datatype
of the subbasin ID field

            if subIDtype != "String":                                       # Convert subbasin
IDs to a string, if they are not already
                subbasinID = [str(i) for i in subbasinID]

            subNameLen = max(map(len, subbasinID))                          # Define the length
of the longest subbasin ID

            # arcpy.AddMessage("\nList of subbasins...\n" + "\n".join(subbasinID))

            arcpy.AddMessage("\nCreating Subbasin Summary Table...")
            tableName = type + "_PMP_Subbasin_Average" + "_" + outArea
            tablePath = outPath + "\\" + outGDB + "\\" + tableName
            dm.CreateTable(outPath + "\\" + outGDB, tableName)              # Create blank table

            dm.AddField(tablePath, "STORM_TYPE", "TEXT", "", "", 10, "Storm Type")          #
Create "Storm Type" field
            dm.AddField(tablePath, "SUBBASIN", "TEXT", "", "", subNameLen, "Subbasin")      #
Create "Subbasin" field

            cursor = arcpy.da.InsertCursor(tablePath, "SUBBASIN")           # Create Insert
cursor and add a blank row to the table for each subbasin
            for sub in subbasinID:
                cursor.insertRow([sub])
            del cursor, sub

            dm.CalculateField(tablePath, "STORM_TYPE", "'" + type + "'", "PYTHON_9.3")      #
populate storm type field

            i = 0
            for field in arcpy.ListFields(pmpPoints, "PMP_*"):              # Add fields for each
PMP duration and calculate the subbasin averages
                fieldName = field.name
                arcpy.AddMessage("\n\tCalculating subbasin average for " + fieldName + "
(weighted)...\n")
                dm.AddField(tablePath, fieldName, "DOUBLE", "", 2)          # Add duration field
                subAveList = []
                for subbasin in subbasinID:                                 # Loop through each
subbasin
                    if subIDtype != "String":                               # Define an SQL
expression that specifies the current subbasin
                        sql_exp = """{0} = {1}""".format(arcpy.AddFieldDelimiters(basin,
subbasinIDfield), subbasin)
                    else:
                        sql_exp = """{0} = '{1}'""".format(arcpy.AddFieldDelimiters(basin,
subbasinIDfield), subbasin)
                    dm.MakeFeatureLayer(basin, "subbasinLayer", sql_exp)
                    outLayer = outPath + "\\" + outGDB + "\\subbasin_" + str(subbasin)
                    subBasAve = basinAve("subbasinLayer", fieldName)        # Call the basAve()
function passing the subbasin and duration field
                    arcpy.AddMessage("\tSubbasin average for " + str(subbasin) + ":  " +
str(subBasAve) + '"')
                    subAveList.append(subBasAve)                            # Add subbasin
average to list
                p = 0
                with arcpy.da.UpdateCursor(tablePath, fieldName) as cursor: # Update the subbasin
average summary table with the subbasin averages
                    for row in cursor:
                        row = subAveList[p]
                        cursor.updateRow([row])
                        p += 1

    ##            dm.CalculateField(tablePath, fieldName, fieldAve, "PYTHON_9.3")       # Assigns
the basin average
    ##            dur = durList[i]                                                      #
following lines add alias field names to basin average table (ArcGIS 10.2.1 or later)
    ##            if dur[0] == "0":
    ##                dur = dur[1:]
    ##            fieldAlias = dur + "-hour PMP"
    ##            dm.AlterField(tablePath, fieldName, "#", fieldAlias)
                i += 1
            arcpy.AddMessage("\nSubbasin summary table complete.")

        arcpy.AddMessage("\nCreating Basin Summary Table...")
        tableName = type + "_PMP_Basin_Average" + "_" + outArea
        tablePath = outPath + "\\" + outGDB + "\\" + tableName
        dm.CreateTable(outPath + "\\" + outGDB, tableName)          # Create blank table
        cursor = arcpy.da.InsertCursor(tablePath, "*")              # Create Insert cursor and
add a blank row to the table
        cursor.insertRow([0])
        del cursor

        dm.AddField(tablePath, "STORM_TYPE", "TEXT", "", "", 30, "Storm Type")          # Create
"Storm Type" field
        dm.CalculateField(tablePath, "STORM_TYPE", "'" + type + "'", "PYTHON_9.3")      #
populate storm type field

        i = 0
        for field in arcpy.ListFields(pmpPoints, "PMP_*"):          # Add fields for each PMP
duration and calculate the basin average
            fieldName = field.name
            fieldAve = basinAve(basin, fieldName)                   # Calls the basinAve()
function - returns the average (weighted or not)
            dm.AddField(tablePath, fieldName, "DOUBLE", "", 2)      # Add duration field
            dm.CalculateField(tablePath, fieldName, fieldAve, "PYTHON_9.3")       # Assigns the
basin average
##            dur = durList[i]                                      # following lines add alias
field names to basin average table (ArcGIS 10.2.1 or later)
##            if dur[0] == "0":
##                dur = dur[1:]
##            fieldAlias = dur + "-hour PMP"
##            dm.AlterField(tablePath, fieldName, "#", fieldAlias)
            i += 1
        arcpy.AddMessage("\nSummary table complete.")
        basAveTables.append(tablePath)

##             The following lines export a .png image depth duration chart and PMP summary excel
file to the output folder
        if ddChart:
            xValues = durList                       #Get list of durations for chart
            xValues = [int(i) for i in xValues]            #Convert duration list to integers
            ax1 = plt.subplot2grid((1,1), (0,0))    #Create variable for subplot in chart
            yValues = []
            pmpFields = [field.name for field in arcpy.ListFields(tablePath, "PMP_*")] # Selects
PMP fields for yValues
            with arcpy.da.SearchCursor(tablePath, pmpFields) as cursor:               # Adds PMP
depths to yValues
                yValues = next(cursor)
            del cursor, pmpFields

            stormFields = [field.name for field in arcpy.ListFields(pmpPoints, "Storm_*")] #
Selects Controlling Storm fields
            contStorms = []                            # List of controlling storms for a single
duration
            listOfContStorms = []                      # List of controlling storms for all
durations (list of lists)
            i = 0                                      # iterator (for "Storm_*" fields)
            while i < len(stormFields):                     # iterates through controlling storm
fields
                with arcpy.da.SearchCursor(pmpPoints, stormFields) as cursor:    # Search cursor
returns list of unique controlling storms
```

```
                    contStorms = sorted({row[i] for row in cursor})
                listOfContStorms.append(contStorms)                         # Add unique storms
for current duration to list of controlling stomrs
                i += 1
            del cursor

            plt.plot(xValues,yValues)        #Creates chart
            plt.xlabel('Storm Duration in Hours')
            plt.ylabel('Rainfall Depth in Inches')
            plt.title(basinName + " (" + outArea + ") " + type + ' Storm Basin Average PMP\nDepth
Duration Chart')
            ax1.grid(True)                           #Creates grid lines in chart
            yTop = max(yValues) + 1
            ax1.set_ylim(top = yTop)         #Sets y axis values to match depths +1 1
            ax1.set_xticks(xValues)          #Sets x axis values to match durations
##            i = 0
##            xy = zip(xValues, yValues)
##            while i < len(stormFields):                     # iterates through controlling
storm fields
##                pointXY = xy[i]
##                yLabel = '{0:.1f}'.format(yValues[i])       # round PMP depth to 1 decimal and
convert to string
##                stormLabel = str(listOfContStorms[i])       # convert controlling storm ID(s)
to string
##                stormLabel = stormLabel.replace("u", "")    # remove unicode "u"
##                stormLabel = stormLabel.replace("'", "")    # remove unicode ","
##                stormLabel = stormLabel.replace("[", "")    # remove unicode "["
##                stormLabel = stormLabel.replace("]", "")    # remove unicode "]"
##                #ax1.annotate(yLabel + '"\n' + stormLabel, xy=xy[i], textcoords='offset
points', size=8, annotation_clip=True)
##                ax1.annotate(yLabel + '"\n' + stormLabel, xy=xy[i], textcoords='data', size=8,
annotation_clip=True)
##                i += 1
##            del xy

            plt.savefig(outPath + "\\" + basinName + "_" + type + "_Depth_Duration_Chart.png")
        #Save image
            plt.close()                      #Close chart to remove from memory
            arcpy.AddMessage("\nDepth Duration Chart exported to output folder.")
            del xValues, yValues, #df, dfLimited
            return
        return

    ###########################################################################
    ##  The basin() returns the basin average PMP value for a given duration field.
    ##  If the option for a weighted average is checked in the tool parameter the script
    ##  will weight the grid point values based on proportion of area inside the basin.
    def basinAve(aoiBasin, pmpField):
        pmpPoints = env.scratchGDB + "\\PMP_Points"
# Path of 'PMP_Points' scratch feature class
        if weightedAve:
            #arcpy.AddMessage("\tCalculating sub-basin average for " + pmpField +
"(weighted)...")
            vectorGridClip = env.scratchGDB + "\\VectorGridClip"
# Path of 'VectorGridClip' scratch feature class

            dm.MakeFeatureLayer(home + "\\Input\\Non_Storm_Data.gdb\\Vector_Grid", "vgLayer")
# make a feature layer of vector grid cells
            dm.SelectLayerByLocation("vgLayer", "INTERSECT", aoiBasin)
# select the vector grid cells that intersect the aoiBasin polygon

            an.Clip("vgLayer", aoiBasin, vectorGridClip)
# clips aoi vector grid to basin
            dm.AddField(pmpPoints, "WEIGHT", "DOUBLE")
# adds 'WEIGHT' field to PMP_Points scratch feature class
            dm.MakeFeatureLayer(vectorGridClip, "vgClipLayer")
# make a feature layer of basin clipped vector grid cells
            dm.MakeFeatureLayer(pmpPoints, "pmpPointsLayer")
# make a feature layer of PMP_Points feature class
```

```
            dm.AddJoin("pmpPointsLayer", "ID", "vgClipLayer", "ID")
# joins PMP_Points and vectorGridBasin tables
            dm.CalculateField("pmpPointsLayer", "WEIGHT", "!vectorGridClip.Shape_Area!",
"PYTHON_9.3")      # Calculates basin area proportion to use as weight for each grid cell.
            dm.RemoveJoin("pmpPointsLayer", "vectorGridClip")

            dm.SelectLayerByLocation("pmpPointsLayer", "INTERSECT", "vgLayer")

            na = arcpy.da.TableToNumPyArray("pmpPointsLayer",(pmpField, 'WEIGHT'))
# Assign pmpPoints values and weights to Numpy array (na)
            wgtAve = np.average(na[pmpField], weights=na['WEIGHT'])
# Calculate weighted average with Numpy average
            del na
            return round(wgtAve, 2)

        else:
            if includeSubbasin:
                #arcpy.AddMessage("\tCalculating sub-basin average for " + pmpField + "(non-
weighted)...")
                vectorGridClip = env.scratchGDB + "\\VectorGridClip"
# Path of 'VectorGridClip' scratch feature class

                dm.MakeFeatureLayer(home + "\\Input\\Non_Storm_Data.gdb\\Vector_Grid", "vgLayer")
# make a feature layer of vector grid cells
                dm.SelectLayerByLocation("vgLayer", "INTERSECT", aoiBasin)
# select the vector grid cells that intersect the aoiBasin polygon

                dm.MakeFeatureLayer(pmpPoints, "pmpPointsLayer")
# make a feature layer of PMP_Points feature class

                dm.SelectLayerByLocation("pmpPointsLayer", "INTERSECT", "vgLayer")

                na = arcpy.da.TableToNumPyArray("pmpPointsLayer", pmpField)
# Assign pmpPoints values and weights to Numpy array (na)
                fieldAve = np.average(na[pmpField])
# Calculates aritmetic mean
                del na
                return round(fieldAve, 2)

            else:
                arcpy.AddMessage("\tCalculating basin average for " + pmpField + "(not
weighted)...")
                na = arcpy.da.TableToNumPyArray(pmpPoints, pmpField)
# Assign pmpPoints values to Numpy array (na)
                fieldAve = np.average(na[pmpField])
# Calculates aritmetic mean
                del na
                return round(fieldAve, 2)

    ###########################################################################
    ##  This basinZone() function returns a list containing transposition zone ID
    ##  (as an integer) and side of continental divide of the the basin centroid
    ##  (as text "East" or "West")
    def basinZone(bas):        ## This function returns the transposition zone and side of
continental divide of the the basin centroid
        tempBasin = env.scratchGDB + "\\tempBasin"
        tempCentroid = env.scratchGDB + "\\tempCentroid"
        joinFeat = home + "\\Input\\Non_Storm_Data.gdb\\Vector_Grid"
        joinOutput = env.scratchGDB + "\\joinOut"
        dm.Dissolve(bas, tempBasin)
        desc = arcpy.Describe(tempBasin)
        sr = desc.spatialReference
        #dm.FeatureToPoint(tempBasin, tempCentroid, "INSIDE")

        dm.CreateFeatureclass(env.scratchGDB,"tempCentroid","POINT",spatial_reference = sr)
        with arcpy.da.InsertCursor(tempCentroid, "SHAPE@XY") as iCur:
            with arcpy.da.SearchCursor(tempBasin,"SHAPE@") as sCur:
                for sRow in sCur:
                    cent = sRow[0].centroid          # get the centroid
                    iCur.insertRow([(cent.X,cent.Y)])# write it to the new feature class
        an.SpatialJoin(tempCentroid, joinFeat, joinOutput)
        centZone = arcpy.da.SearchCursor(joinOutput, ("TRANS_ZONE",)).next()[0]
        centDivide = arcpy.da.SearchCursor(joinOutput, ("DIVIDE",)).next()[0]
        del tempBasin, tempCentroid, joinFeat, joinOutput, desc, sr
        return (centZone, centDivide)

    ##############################################################################
    ##  This portion of the code checks to make sure none of the temporal distributions
    ##  are exceeding the PMP values for any durations.  It adds a table to the output
    ##  folder called CheckTemporal.

##~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

    def checkTemporal(stormType, outPath, TemporalTable, distributionFields, dur, areaSize):
        basinPMP = outPath + "\\" + stormType + "_PMP_Basin_Average_" + areaSize
# Location of basin average PMP table
        pmpFields = [field.name for field in arcpy.ListFields(basinPMP, "PMP_*")]
# PMP duration run

        pmp = []
#Creates empty list and updates with PMP values for each duration run
        i = 0
        while i < len(pmpFields):
            with arcpy.da.SearchCursor(basinPMP,pmpFields) as cursor:
                for row in cursor:
                    pmp.append(row[i])
                    i += 1
        del i, cursor

        checkTable = arcpy.CreateTable_management(outPath, "CheckTemporal_" + dur + "hr")
#Creates table in output folder, adds field, and populates field with distributions
        dm.AddField(checkTable, "PATTERN", "TEXT", "", "", 50)
        with arcpy.da.InsertCursor(checkTable, "PATTERN") as cursor:
            for val in distributionFields:
                cursor.insertRow([val])

        maxFields = []
#Create Max fields for each duration
        for maxField in pmpFields:
            newField = maxField.replace("PMP","MAX")
            maxFields.append(newField)
        del newField

        checkFields = []
#Create Check fields for each duration
        for checkField in pmpFields:
            newField = checkField.replace("PMP","CHECK")
            checkFields.append(newField)
        del newField

        i = 0
#Populate fields
        for pmpField in pmpFields:
            dm.AddField(checkTable, pmpField, "DOUBLE", "", "", 50)
            dm.AddField(checkTable, maxFields[i], "DOUBLE", "", "", 50)
            dm.AddField(checkTable, checkFields[i], "TEXT", "", "", 50)
            with arcpy.da.UpdateCursor(checkTable, pmpField) as cursor:
                for row in cursor:
                    row = pmp[i]
                    cursor.updateRow([row])
                i += 1
        del i, cursor

        dic = {"01": 4, "02": 8, "03": 12, "04": 16, "05": 20, "06": 24, "12": 48, "24": 96,
"48": 192, "72": 288, "96": 384, "120": 480}  # Dictionary to convert durations into 15-minute
timesteps

        i = 0
# Calculates incremental PMP depths from temporal distribution and gets maximum rainfall for each
duration run
```

```
        for dur in durList:
            k = dic[dur]
            maxpmpList = []
            p = 0
            for distribution in distributionFields:
                incPMP = []
                previousRow = 0
                with arcpy.da.SearchCursor(TemporalTable, distributionFields) as cursor:
                    for row in cursor:
                        increment = row[p] - previousRow
                        previousRow = row[p]
                        incPMP.append(increment)
                na = np.array(incPMP)
                sumList = np.convolve(na,np.ones(k))
                maxPMP = max(sumList)
                maxpmpList.append(maxPMP)
                p +=1
            x = 0
            with arcpy.da.UpdateCursor(checkTable, maxFields[i]) as cursor:
# Updates table with max values
                for row in cursor:
                    row = maxpmpList[x]
                    x += 1
                    cursor.updateRow([row])
            i += 1
        del i, k, cursor, x

        with arcpy.da.UpdateCursor(checkTable, '*') as cursor:
# Compares PMP values to max values for each duration.  If PMP values are larger update check
field with PASS if not FAIL
            for row in cursor:
                rec = dict(zip(cursor.fields, row))
                arcpy.AddMessage("\n\tChecking values for distribution....." + rec['PATTERN'])
                for k, v in rec.iteritems():
                    if not k.startswith('PMP_'):
                        continue
                    _, n = k.split('_')
                    mx = rec['MAX_{}'.format(n)]
                    rec['CHECK_{}'.format(n)] = 'FAIL' if v < mx else 'PASS'
                    if rec['CHECK_{}'.format(n)] == 'PASS':
                        arcpy.AddMessage(str(k) + " \n\tPMP value is... " + str(v) + "  \n\tmax
rainfall value is..." + str(mx) + "\n\tThis distribution.... " + rec['CHECK_{}'.format(n)])
                    else:
                        arcpy.AddMessage(str(k) + " \n\tPMP value is... " + str(v) + "  \n\tmax
rainfall value is..." + str(mx) + "\n\tThis distribution.... " + rec['CHECK_{}'.format(n)]+ "\n\t
Do not use this distribution.  Max values for duration are exceeding PMP values.  Use critically
stacked distribution.... ")
                cursor.updateRow([rec[k] for k in cursor.fields])
        del cursor, k, v, rec
        return

    ###########################################################################
    ##  The temporalDistLS() function applies the temporal distributions scenarios
    ##  to local storm PMP
    def temporalDistLS(stormType, outPath, areaSize):
               # Local Storm Temporal Distributions Application Function
       basinPMP = outPath + "\\" + stormType + "_PMP_Basin_Average_" + areaSize
# Location of basin average PMP table

        arcpy.AddMessage("\n***Local Storm - 2-hour PMP Temporal Distributions***")

        temporalDistTable = home + "\\Input\\Non_Storm_Data.gdb\\LS_Temporal_Distributions_2hr"
# LS 2hr Temporal distribution factors tables
        distributionList = [field.name for field in arcpy.ListFields(temporalDistTable,
"LS_2_*")]            # Create a list of 2-hour distribution field names
        outTable = outPath + "\\LS_Temporal_Distributions_2hr"
        arcpy.AddMessage("\n\tCreating 2-hour temporal distribution table:...")
        dm.Copy(temporalDistTable, outTable)
# Copy 2-hour temporal dist. factors table to output location
        largestHour = arcpy.da.SearchCursor(basinPMP, ("PMP_01",)).next()[0]
               # Calculate largest hour PMP
        secondLargestHour = arcpy.da.SearchCursor(basinPMP, ("PMP_02",)).next()[0] - largestHour
        # Calculate 2nd-largest hour PMP
        arcpy.AddMessage("\n\tLargest Hour: " + str(largestHour))
        arcpy.AddMessage("\tSecond Largest Hour: " + str(secondLargestHour))
        for distribution in distributionList:
               # Loops thourgh each 2-hour temporal distribution
            arcpy.AddMessage("\n\tApplying temporal distribution for: " + distribution)
            arcpy.AddMessage("\t\tFirst hour...")
            with arcpy.da.UpdateCursor(outTable, distribution) as cursor:
        # Cursor to apply temporal factor to 1-hour PMP
                for row in cursor:
                    if not row[0]:
                        #arcpy.AddMessage("\tLeaving loop...")
                        break
                    #arcpy.AddMessage("\tAccumulated Rain: " + str(round(row[0] *
largestHour,3)))
                    row[0] = round(row[0] * largestHour,3)
                    cursor.updateRow(row)
                del row, cursor
            arcpy.AddMessage("\t\tSecond hour...")
            accumPMP = largestHour
            whereClause = distribution + " IS NULL"
            with arcpy.da.UpdateCursor(outTable, distribution, whereClause) as cursor:
               # Cursor to evenly distribute 2nd largest hour
                for row in cursor:
                    accumPMP += secondLargestHour / 12
                    #arcpy.AddMessage("\tAccumulated Rain: " + str(round(accumPMP, 3)))
                    row[0] = round(accumPMP, 3)
                    cursor.updateRow(row)
                del row, cursor, accumPMP, whereClause
        del distribution, distributionList, largestHour, secondLargestHour

        arcpy.AddMessage("\n***Local Storm - 6-hour PMP Temporal Distributions***")

        temporalDistTable = home + "\\Input\\Non_Storm_Data.gdb\\LS_Temporal_Distributions_6hr"
# LS 6hr Temporal distribution factors table
        outTable = outPath + "\\LS_Temporal_Distributions_6hr"
        arcpy.AddMessage("\n\tCreating temporal distribution table:...")
        dm.Copy(temporalDistTable, outTable)
# Copy 6-hour temporal dist. factors table to output location
        distributionList = [field.name for field in arcpy.ListFields(temporalDistTable,
"LS_6_*")]            # Create a list of 6-hour distribution field names
        arcpy.AddMessage("\n\tDistribution Field Names: " + str(distributionList))
        sixHour = arcpy.da.SearchCursor(basinPMP, ("PMP_06",)).next()[0]
        # Gets 6-hour PMP depth
        for distribution in distributionList:
               # Loops thourgh each 6-hour temporal distribution
            arcpy.AddMessage("\n\tApplying temporal distribution for: " + distribution)
            with arcpy.da.UpdateCursor(outTable, distribution) as cursor:
        # Cursor to apply temporal factor to 6-hour PMP
                for row in cursor:
                    #arcpy.AddMessage("\tAccumulated Rain: " + str(round(row[0] * sixHour, 3)))
                    row[0] = round(row[0] * sixHour, 3)
                    cursor.updateRow(row)
                del row, cursor

        #location[0] == 1 or location[0] == 2       or location[0] == 3 or location[0] == 12 or
location[0] == 13:    # condition if basin centroid is in one of the Hybrid zones
        arcpy.AddMessage("\n***Local Storm - 24-hour Hybrid PMP Temporal Distributions***")

        temporalDistTable = home +
"\\Input\\Non_Storm_Data.gdb\\Hybrid_Temporal_Distributions_24hr"          # 24-hour Temporal
distribution factors table
        distributionList = [field.name for field in arcpy.ListFields(temporalDistTable,
"LS_24_*")]            # Create a list of 24-hour distribution field names
        arcpy.AddMessage("\n\tDistribution Field Names: " + str(distributionList))
        outTable = outPath + "\\Hybrid_Temporal_Distributions_24hr"
        arcpy.AddMessage("\n\tCreating temporal distribution table:...")
        dm.Copy(temporalDistTable, outTable)
# Copy 24-hour temporal dist. factors table to output location
```

```
        twentyfourHour = arcpy.da.SearchCursor(basinPMP, ("PMP_24",)).next()[0]
                # Gets 24-hour PMP depth
        for distribution in distributionList:
                # Loops thourgh each 24-hour temporal distribution
            arcpy.AddMessage("\n\tApplying temporal distribution for: " + distribution)
            with arcpy.da.UpdateCursor(outTable, distribution) as cursor:
       # Cursor to apply temporal factor to 24-hour PMP
                for row in cursor:
                    #arcpy.AddMessage("\tAccumulated Rain: " + str(round(row[0] * twentyfourHour,
3)))
                    row[0] = round(row[0] * twentyfourHour, 3)
                    cursor.updateRow(row)
                del row, cursor
        distType = "24hr_Hybrid"
        checkTemporal(stormType, outPath, outTable, distributionList, distType, areaSize)
        del basinPMP, distribution, distributionList, outTable, temporalDistTable
        return

    ############################################################################
    ##  The temporalDistGS() function applies the temporal distributions scenarios
    ##  to general or tropical storm PMP based on the basin location.  The function
    ##  accepts the storm type as the first argument, the basin average table location
    ##  as the second argument
    def temporalDistGS(stormType, outPath, areaSize):
                # General Storm Temporal Distributions Application Function
       basinPMP = outPath + "\\" + stormType + "_PMP_Basin_Average_" + areaSize
# Location of basin average PMP table

        arcpy.AddMessage("\n***" + stormType + " Storm - PMP Temporal Distributions***")

        temporalDistTable = home + "\\Input\\Non_Storm_Data.gdb\\GS_Temporal_Distributions"
                # General Storm Temporal distribution factors table
        distributionList = [field.name for field in arcpy.ListFields(temporalDistTable, "GS*")]
               # Create a list of 24-hour distribution field names
        arcpy.AddMessage("\n\tDistribution Field Names: " + str(distributionList))
        outTable = outPath + "\\GS_Temporal_Distributions"

        arcpy.AddMessage("\n\tCreating temporal distribution table:...")
        dm.Copy(temporalDistTable, outTable)
       # Copy temporal dist. factors table to output location

        largest24 = arcpy.da.SearchCursor(basinPMP, ("PMP_24",)).next()[0]
                                                                                            #
Calculate largest 24-hour period PMP
        second24 = arcpy.da.SearchCursor(basinPMP, ("PMP_48",)).next()[0] - largest24
                # Calculate 2nd-largest 24-hour period PMP
        third24 = arcpy.da.SearchCursor(basinPMP, ("PMP_72",)).next()[0] -
arcpy.da.SearchCursor(basinPMP, ("PMP_48",)).next()[0]      # Calculate 3rd-largest 24-hour
period PMP

        arcpy.AddMessage("\n\tLargest 24-hour: " + str(largest24))
        arcpy.AddMessage("\tSecond largest 24-hour: " + str(second24))
        arcpy.AddMessage("\tThird largest 24-hour: " + str(third24))

        for distribution in distributionList:
               # Loops thourgh each 24-hour temporal distribution
            arcpy.AddMessage("\n\tApplying temporal distribution for: " + distribution)
            arcpy.AddMessage("\t\tFirst 24-hour Period...")
            accumPMP = 0
            with arcpy.da.UpdateCursor(outTable, [distribution, "TIMESTEP"]) as cursor:
               # Cursor to evenly distribute 2nd largest hour
                for row in cursor:
                    if row[1] <= 96:
       # Leave loop once a row containing a temporal dist. factor (ie, second 24h period) is
reached
                        accumPMP +=  second24 / 96
                        #arcpy.AddMessage("\tAccumulated Rain: " + str(round(accumPMP, 3)))
                        row[0] = round(accumPMP, 3)
                        cursor.updateRow(row)
                del row, cursor
            arcpy.AddMessage("\t\tSecond 24-hour Period...")
            with arcpy.da.UpdateCursor(outTable, [distribution, "TIMESTEP"]) as cursor:
               # Cursor to apply temporal factors to largest 24-hour PMP
                for row in cursor:
                    if row[1] > 96 and row[1] <= 192:
               # Constrain update to rows 97-192 (second 24hr period)
                        accumPMP = round((largest24 * row[0]) + second24, 3)
                        #arcpy.AddMessage("\tAccumulated Rain: " + str(accumPMP))
                        row[0] = accumPMP
                        cursor.updateRow(row)
                del row, cursor

            arcpy.AddMessage("\t\tThird 24-hour Period...")
            whereClause = distribution + " IS NULL"
            with arcpy.da.UpdateCursor(outTable, distribution, whereClause) as cursor:
               # Cursor to evenly distribute 3nd largest hour over remaining empty rows
                for row in cursor:
                    accumPMP +=  third24 / 96
                    #arcpy.AddMessage("\tAccumulated Rain: " + str(round(accumPMP, 3)))
                    row[0] = round(accumPMP, 3)
                    cursor.updateRow(row)
                del row, cursor, accumPMP, whereClause
            #arcpy.AddMessage("\nCompleted temporal distribution for: " + distribution)

        distType = "24hr_GS"
        checkTemporal(stormType, outPath, outTable, distributionList, distType, areaSize)
        del stormType, basinPMP
        return

    ###########################################################################
    ##  The temporalDistTS() function applies the temporal distributions scenarios
    ##  to general or tropical storm PMP based on the basin location.  The function
    ##  accepts the storm type as the first argument, the basin average table location
    ##  as the second argument
    def temporalDistTS(stormType, outPath, areaSize):
                # Tropical Storm Temporal Distributions Application Function
       basinPMP = outPath + "\\" + stormType + "_PMP_Basin_Average_" + areaSize
# Location of basin average PMP table

        arcpy.AddMessage("\n***" + stormType + " Storm - PMP Temporal Distributions***")

        temporalDistTable = home + "\\Input\\Non_Storm_Data.gdb\\TS_Temporal_Distributions"
       # Tropical Storm Temporal distribution factors table
        distributionList = [field.name for field in arcpy.ListFields(temporalDistTable, "TS*")]
               # Create a list of 24-hour distribution field names
        arcpy.AddMessage("\n\tDistribution Field Names: " + str(distributionList))
        outTable = outPath + "\\TS_Temporal_Distributions"
        arcpy.AddMessage("\n\tCreating temporal distribution table:...")
        dm.Copy(temporalDistTable, outTable)
       # Copy temporal dist. factors table to output location

        largest24 = arcpy.da.SearchCursor(basinPMP, ("PMP_24",)).next()[0]
                                                                                           #
Calculate largest 24-hour period PMP
        second24 = arcpy.da.SearchCursor(basinPMP, ("PMP_48",)).next()[0] - largest24
                # Calculate 2nd-largest 24-hour period PMP
        third24 = arcpy.da.SearchCursor(basinPMP, ("PMP_72",)).next()[0] -
arcpy.da.SearchCursor(basinPMP, ("PMP_48",)).next()[0]      # Calculate 3rd-largest 24-hour
period PMP

        arcpy.AddMessage("\n\tLargest 24-hour: " + str(largest24))
        arcpy.AddMessage("\tSecond largest 24-hour: " + str(second24))
        arcpy.AddMessage("\tThird largest 24-hour: " + str(third24))

        for distribution in distributionList:
               # Loops thourgh each 24-hour temporal distribution
            arcpy.AddMessage("\n\tApplying temporal distribution for: " + distribution)
            arcpy.AddMessage("\t\tFirst 24-hour Period...")
            accumPMP = 0
            with arcpy.da.UpdateCursor(outTable, [distribution, "TIMESTEP"]) as cursor:
               # Cursor to evenly distribute 2nd largest hour
```

```
                for row in cursor:
                    if row[1] <= 96:
        # Leave loop once a row containing a temporal dist. factor (ie, second 24h period) is
reached
                        accumPMP +=  second24 / 96
                        #arcpy.AddMessage("\tAccumulated Rain: " + str(round(accumPMP, 3)))
                        row[0] = round(accumPMP, 3)
                        cursor.updateRow(row)
                del row, cursor

            arcpy.AddMessage("\t\tSecond 24-hour Period...")
            with arcpy.da.UpdateCursor(outTable, [distribution, "TIMESTEP"]) as cursor:
               # Cursor to apply temporal factors to largest 24-hour PMP
                for row in cursor:
                    if row[1] > 96 and row[1] <= 192:
               # Constrain update to rows 97-192 (second 24hr period)
                        accumPMP = round((largest24 * row[0]) + second24, 3)
                        #arcpy.AddMessage("\tAccumulated Rain: " + str(accumPMP))
                        row[0] = accumPMP
                        cursor.updateRow(row)
                del row, cursor

            arcpy.AddMessage("\t\tThird 24-hour Period...")
            whereClause = distribution + " IS NULL"
            with arcpy.da.UpdateCursor(outTable, distribution, whereClause) as cursor:
               # Cursor to evenly distribute 3nd largest hour over remaining empty rows
                for row in cursor:
                    accumPMP +=  third24 / 96
                    #arcpy.AddMessage("\tAccumulated Rain: " + str(round(accumPMP, 3)))
                    row[0] = round(accumPMP, 3)
                    cursor.updateRow(row)
                del row, cursor, accumPMP, whereClause
            #arcpy.AddMessage("\nCompleted temporal distribution for: " + distribution)

        distType = "24hr_TS"
        checkTemporal(stormType, outPath, outTable, distributionList, distType, areaSize)
        del stormType, basinPMP
        return

    ###########################################################################
    ##  This portion of the code iterates through each storm feature class in the
    ##  'Storm_Adj_Factors' geodatabase (evaluating the feature class only within
    ##  the Local, Tropical, or general feature dataset).  For each duration,
    ##  at each grid point within the aoi basin, the transpositionality is
    ##  confirmed.  Then the DAD precip depth is retrieved and applied to the
    ##  total adjustement factor to yield the total adjusted rainfall.  This
    ##  value is then sent to the updatePMP() function to update the 'PMP_Points'
    ##  feature class.
##~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~##

    desc = arcpy.Describe(basin)                                                        # Check
to ensure AOI input shape is a Polygon. If not - exit.
    basinShape = desc.shapeType
    if desc.shapeType == "Polygon":
        arcpy.AddMessage("\nBasin shape type: " + desc.shapeType)
    else:
        arcpy.AddMessage("\nBasin shape type: " + desc.shapeType)
        arcpy.AddMessage("\nError: Input shapefile must be a polygon!\n")
        sys.exit()

    createPMPfc()                                                                       # Call
the createPMPfc() function to create the PMP_Points feature class.

    env.workspace = adjFactGDB                                                          # the
workspace environment is set to the 'Storm_Adj_Factors' file geodatabase

    aoiSQMI = round(getAOIarea(),2)                                                     # Calls
the getAOIarea() function to assign area of AOI shapefile to 'aoiSQMI'
    if aoiSQMI > 100 and stormType is "Local":
```

```
        arcpy.AddMessage("\n***Warning - Local storm PMP depths only valid for basins 100 square
miles or smaller***")

    stormList = arcpy.ListFeatureClasses("", "Point", stormType)                        # List
all the total adjustment factor feature classes within the storm type feature dataset.
    for dur in durList:

arcpy.AddMessage("\n*************************************************************\nEvaluating " +
dur + "-hour duration...")

        pmpList = []
        driverList = []
        gridRows = arcpy.SearchCursor(env.scratchGDB + "\\PMP_Points")
        try:
            for row in gridRows:
                pmpList.append(0.0)                                                         #
creates pmpList of empty float values for each grid point to store final PMP values
                driverList.append("STORM")                                                  #
creates driverList of empty text values for each grid point to store final Driver Storm IDs
            del row, gridRows
        except UnboundLocalError:
            arcpy.AddMessage("\n***Error: No data present within basin/AOI area.***\n")
            sys.exit()

        for storm in stormList[:]:
            arcpy.AddMessage("\n\tEvaluating storm: " + storm + "...")
            dm.MakeFeatureLayer(storm, "stormLayer")                                    # creates
a feature layer for the current storm
            dm.SelectLayerByLocation("stormLayer", "HAVE_THEIR_CENTER_IN", "vgLayer")   #
examines only the grid points that lie within the AOI
            gridRows = arcpy.SearchCursor("stormLayer")
            pmpField = "PMP_" + dur
            i = 0
            try:
                dadPrecip = round(dadLookup(storm, dur, aoiSQMI),3)
                arcpy.AddMessage("\t\t" + dur + "-hour DAD value:  " + str(dadPrecip) + chr(34))
            except TypeError:                                                           # In no
duration exists in the DAD table - move to the next storm
                arcpy.AddMessage("\t***Duration '" + str(dur) + "-hour' is not present for " +
str(storm) + ".***\n")
                continue
            arcpy.AddMessage("\t\tComparing " + storm + " adjusted rainfall values against
current driver values...")
            transCounter = 0                                                    # Counter for
number of grid points transposed to
            for row in gridRows:
                if row.TRANS == 1:                                              # Only continue
if grid point is transpositionable ('1' is transpostionable, '0' is not).
                    try:                                                        # get total adj.
factor if duration exists
                        transCounter += 1
                        adjRain = round(dadPrecip * row.TAF,1)
                        if adjRain > pmpList[i]:
                            pmpList[i] = adjRain
                            driverList[i] = storm
                    except RuntimeError:
                        arcpy.AddMessage("\t\t   *Warning*  Total Adjusted Raifnall value falied
to set for row " + str(row.CNT))
                        break
                    del adjRain
                i += 1
            if transCounter == 0:
                arcpy.AddMessage("\t\tStorm not transposable to basin. Removing " + storm + "
from list...\n")
                stormList.remove(storm)
            else:
                arcpy.AddMessage("\t\tTransposed to " + str(transCounter) + "/" + str(i) + " grid
points...\n")
            del row, transCounter
        del storm, gridRows, dadPrecip
        updatePMP(pmpList, driverList, dur)              # calls function to update "PMP Points" 
feature class
    del dur, pmpList, stormList

    arcpy.AddMessage("\n'PMP_Points' Feature Class 'PMP_XX' fields update complete for all '" + 
stormType + "' storms.")

    outputPMP(stormType, aoiSQMI, outputPath)               # calls outputPMP() function

    if runTemporal:
        # centroidLocation = basinZone(basin)
        # arcpy.AddMessage("\nBasin Centroid Transposition Zone: " + str(centroidLocation[0]) + 
"\nBasin Centroid side of Continental Divide: " + str(centroidLocation[1]))
        outArea = str(int(round(aoiSQMI,0))) + "sqmi"
        outGDB = outLocation + "\\" + stormType + "\\PMP_" + desc.baseName + "_" + outArea + 
".gdb"
        if stormType == "Local":
            temporalDistLS(stormType, outGDB, outArea)
        if stormType == "General":
            temporalDistGS(stormType, outGDB, outArea)
        if stormType == "Tropical":
            temporalDistTS(stormType, outGDB, outArea)
        del outGDB

    del aoiSQMI, stormType
    return
##~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~##

def outputBasAveTable():
    arcpy.AddMessage("\nCreating basin average summary table.\n")
    tableList = basAveTables
    for table in tableList:
        arcpy.AddMessage("\t\tMerging tables... " + table)

    dm.Merge(basAveTables, outputTable)
    ##  addLayerMXD(outputTable)  adds output table to ArcMap session

    return
##~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~##

def addLayerMXD(addFC):
    desc = arcpy.Describe(addFC)
    layerName = desc.name
    arcpy.AddMessage("\nAdding " + layerName + " table to current MXD...")
    if desc.dataType == "FeatureClass":
        dm.MakeFeatureLayer(addFC, layerName)
        layer = arcpy.mapping.Layer(layerName)
        arcpy.mapping.AddLayer(df, layer)
        arcpy.AddMessage("\n" + layerName + " added to current map session.\n")
    elif desc.dataType == "Table":
        layer = arcpy.mapping.TableView(desc.catalogPath)
        arcpy.mapping.AddTableView(df, layer)
        arcpy.AddMessage("\n" + layerName + " added to current map session.\n")
    elif desc.dataType == "ArcInfoTable":
        layer = arcpy.mapping.TableView(desc.catalogPath + ".dbf")
        arcpy.mapping.AddTableView(df, layer)
        arcpy.AddMessage("\n" + layerName + " added to current map session.\n")

    del desc, layerName, layer
    return

##~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~##

if locDurations:
    type = "Local"
    durations = locDurations
    dm.CreateFolder(outLocation, type)
    outputPath = outLocation + "\\Local\\"
```

```
    arcpy.AddMessage("\nRunning PMP analysis for storm type: " + type)
    pmpAnalysis(basin, type, durations)          # Calls the pmpAnalysis() function to calculate 
the local storm PMP
    #summaryReport()
    arcpy.AddMessage("\nLocal storm analysis 
complete...\n************************************************************************************
*********************")

if genDurations:
    type = "General"
    durations = genDurations
    dm.CreateFolder(outLocation, type)
    outputPath = outLocation + "\\General\\"
    arcpy.AddMessage("\nRunning PMP analysis for storm type: " + type)
    pmpAnalysis(basin, type, durations)          # Calls the pmpAnalysis() function to calculate 
the general storm PMP
    arcpy.AddMessage("\nGeneral Winter storm analysis 
complete...\n************************************************************************************
*********************")

if tropDurations:
    type = "Tropical"
    durations = tropDurations
    dm.CreateFolder(outLocation, type)
    outputPath = outLocation + "\\Tropical\\"
    arcpy.AddMessage("\nRunning PMP analysis for storm type: " + type)
    pmpAnalysis(basin, type, durations)          # Calls the pmpAnalysis() function to calculate 
the tropical storm PMP
    arcpy.AddMessage("\nTropical storm analysis 
complete...\n************************************************************************************
*********************")
```

# Appendix I

## PMP Version Log: Changes to Storm Database and Adjustment Factors

## PMP Version 1.1:

The purpose of this version was to apply an preliminary set of changes to the initial transposition conditions.

### General Storms:

- SPAS 1208_1 – Warner Park, TN 2010 - Added zones 8, 11, and 12 for locations below 3,000' for continuity over OK
- SPAS 1242_1 – Alley Spring, MO 2008 - Added zones 8, 11, and 12 for locations below 3,000' for
- SPAS 1286_1 – Aurora College, IL 1996 – Updated DAD table with log-linear DAD values
- SPAS 1305_1 - Elba, AL 1929 – Limit to +/- 5° Latitude (36.363°N)
- SPAS 1583_1 – Council Grove, KS 1951 – Limit to west of 91°W
- SPAS 1587_1 – Prairieview, NM 1941 – Removed elevation constraint
- SPAS_1591_1 - Hearne, TX 1899 – Constrained to south of 35.84°N, within 5° latitude of storm center

### Hybrid Storms:

The GTF was revised for the hybrid events to use only a single duration (6hr or 24hr) to represent both storm types so that the geographic adjustment would be consistent regardless of PMP type.

- SPAS 1293_3 – Elbert, CO 1965 – Used 6-hour 100-year precipitation estimates for both LS and GS
- SPAS 1294_1 – Penrose, CO 1921 – Used 6-hour 100-year precipitation estimates for both LS and GS
- SPAS 1294_2 – Adelaide, CO 1921 – Used 6-hour 100-year precipitation estimates for both LS and GS
- SPAS 1560_1 – Conway, TX 1951 – Used 24-hour 100-year precipitation estimates for both LS and GS
- SPAS 1568_1 – Carlsbad, NM 1966 – Used 24-hour 100-year precipitation estimates for both LS and GS
- SPAS 1592_1 – Thrall, TX 1921 – Used 24-hour 100-year precipitation estimates for both LS and TS; Constrained to south of 35.629°N, within 5° latitude of storm center. Capped GTF at 1.20. Limit elevation to +/- 1,000' of storm center
- SPAS 1602_1 – Vic Pierce, TX 1954 – Used 24-hour 100-year precipitation estimates for both LS and TS; Restricted south of 36°N and west of 96°W

### Local Storms:

- SPAS 1434_1 – Holt, MO 1947 – Extended the southern limit from 34°N to 33°N to provide a smoother transition
- SPAS 1494_1 – Mountain Home, TX 1932 – Added zones 11 and 12 to match SPAS 1602 limits; Restricted to south of 36°N and 96°W
- SPAS 1495_1 – Cheyenne, OK 1934 – Applied a +/- 1,000' elevation constraint

**Tropical Storms:**
Applied a GTF cap of 1.10 to all tropical events for consistency with the Texas PMP study in addition to the following changes

- SPAS 1631_1 – Watson, LA 2016 – Added this event
- SPAS 1631_2 – Lafayette, LA 2016 – Added this event
- SPAS 1667_1 - Hurricane Harvey, 2017 – Revised DAD table and storm adjustment factors with final SPAS output
- For the following coastal tropical events, a limit of zone 5 and 9 with a buffer of ~3° latitude inland form coast was applied:
    - SPAS 1463
    - SPAS 1569
    - SPAS 1593
    - SPAS 1596
    - SPAS 1601 (zone 1 and 2)
    - SPAS 1667

## PMP Version 1.2:

**General Storms:**

- SPAS 1208_1 – Warner Park, TN 2010 – Added zone 5 and extent to an elevation of 3,300' to provide a smoother field over LA
- SPAS 1218_1 – Douglasville, GA 2009 – Added zone 5 to provide a smoother field over LA
- SPAS 1591_1 – Hearne, TX 1899 – Extended north to 36°N

## PMP Version 1.2a:

The purpose of this version was to evaluate restricting the transposition limits of the storms occurring over the Edwards Plateau region of Texas to only transposition zone 8.

The following tropical storm was constrained to zone 8:

- SPAS 1582 Zone 1 – Broome, TX 1936 (A version was run with and without holding this storm to zone 8)

The following tropical/local hybrid storm was constrained to zone 8:

- SPAS 1602 Zone 1 – Vic Pierce, TX 1954

The following local storms were constrained to zone 8:

- SPAS 1494 – Mountain Home, TX 1932
- SPAS 1496 – Woodward Ranch, TX 1935

In addition, to rectify some of the discontinuous tropical storm PMP over eastern Texas at larger area-sizes, the following changes were made.

- SPAS 1317 – Americus, GA 1994 – This storm was allowed to go into all of zone 8 and zone 12

- SPAS 1601 DAD zone 2 – Dinero, MX, 1967 – This storm was removed. This storm was re-entered into the database in version 1.2c, 1.3, as removing it had no major impact after changing the limits of SPAS 1317.

## PMP Version 1.2b:

A sensitivity was done where SPAS 1602 Vic Pierce, 1954 was removed as a tropical storm altogether. This version is the same as 1.2a, without Vic Pierce.

## PMP Version 1.2c:

For this sensitivity, SPAS 1591 – Hearne, TX 1899 was reclassified from a general storm to a tropical storm. When removing Hearne from the general storm list, a discontinuity was evident between zone 5 and zone 9. To rectify this, SPAS 1431 – Warner, OK 1943 was allowed to go into all of zone 5.

## PMP Version 1.3:

For this full version, each of the changes made for the v1.2a-1.2c sensitivities were adopted. In addition, numerous other changes were made to version 1.2 as planned. This included removing elevation constraints over zone 9, adjusting transposition limits, correcting the SPAS 1667 storm rep. dew point, and adding the May 1995 general storm. A full log of the changes are included in the Version 1.3 storm list table and are summarized below:

### General Storms:

- SPAS 1183_1 – Egerton, MO 1965 – Removed the elevation constraint over zone 9
- SPAS 1206_1 – Big Rapids, MI 1986 – Removed elevation constraint east of 96°W. Restricted zone 9 to the areas above 1.4 GTF or north of 36°N
- SPAS 1208_1 – Warner Park, TN, 2010 – Included all of zone 1 and 11.
- SPAS 1218_1 - Douglasville, GA, 2009 - Removed the elevation constraint
- SPAS 1219_1 - Big Fork, AR 1982 - Removed the elevation constraint
- SPAS 1286_1 – Aurora College, IL 1996 - Removed elevation constraint east of 96°W. Restricted zone 9 to the areas above 1.1 GTF or north of 36°N
- SPAS 1431_1 – Warner, OK 1943 Included zone 5 and 11
- SPAS 1433_1 – Collinsville, IL 1946 - Removed elevation constraint
- SPAS 1591_1 – Hearne, TX 1899 – Removed, reclassified as a tropical storm
- SPAS 1719_1 – Necaise, LA 1995 – Added storm

### Local Storms:

- SPAS 1295_3 – Hale, CO 1935 – Added all of zone 11
- SPAS 1494_1 – Mountain Home, TX 1932 – Held to zone 8
- SPAS 1495_1 – Cheyenne, OK 1934 – Assigned a manually delineated transposition limit over zone 9, based on topographic relief
- SPAS 1496_1 – Woodward Ranch, TX 1935 – Held to zone 8
- SPAS 1602_1 – Vic Pierce, TX 1954 - Held to zone 8. Recalculated GTF using 6-hour 100 year precipitation

### Local/Tropical Hybrid Storms:

- SPAS 1592_1 – Thrall TX, 1921 – Removed elevation constraint over zone 9

### Tropical Storms:

- SPAS 1317_1 – Americus, GA 1994 – Added all of zones 8, 9, 11, and 12
- SPAS 1582_1 – Broome, TX 1936 – Held to zone 8
- SPAS 1591_1 – Hearne, TX 1899 – Added as a tropical Storm. Removed elevation constraint over zone 9
- SPAS 1602_1 – Vic Pierce, TX 1954 – Removed, reclassified as a local storm
- SPAS 1667_1 – Hurricane Harvey, 2017 – Corrected storm rep. dew point to 86.0° (IPMF – 1.04)

## PMP Version 1.4

### All Storms:

- Updated target 100-year precipitation value over all Texas grid points with the NOAA Atlas 14 vol.11 values and recalculated GTF based on these values.
- Recalculated IPMF using the climatological maximum 100-year dew point from the Feb. 2019 revised AWA grids. While some storms resulted in no change, the IPMF for many storms changed due to this revision. All storms that had changes to the IPMF were identified in the storm list.

### General Storms:

- SPAS 1206_1 – Big Rapids, MI 1986 – Removed storm due to refined transposition limits outside of project domain
- SPAS 1286_1 - Aurora College, IL 1996 – Extended the transpositionability within zone 9 to everywhere with a GTF < 1.12 (from 1.10)

### Local/General Hybrid Storms:

- SPAS 1560_1 – Conway, TX 1951– Corrected transposition limits – added zone 11. Revised GTF using NA14 100y precipitation at storm center location.

### Local Storms:

- SPAS 1185_1 – Corrigan, TX 1994 - Revised GTF using NA14 100y precipitation at storm center location
- SPAS 1226_1 – College Hill, OH 1963 - Extended south to 35.5°N and west to 94.5°W
- SPAS 1485_1 – Las Cruces, NM 1935 – Added storm
- SPAS 1494_1 – Mountain Home, TX 1932 - Revised GTF using NA14 100y precipitation at storm center location
- SPAS 1496_1 – Woodward Ranch, TX 1935 - Revised GTF using NA14 100y precipitation at storm center location
- SPAS 1590_1 – Dawson, TX 2015 - Revised GTF using NA14 100y precipitation at storm center location
- SPAS 1602_1 – Vic Pierce, TX 1954 - Revised GTF using NA14 100y precipitation at storm center location

**Local/Tropical Hybrid Storms:**

- SPAS 1592_1 – Thrall TX, 1921 – Revised GTF using NA14 100y precipitation at storm center location

**Tropical Storms:**

- SPAS 1179_1 – Albany, TX 1978 – Revised GTF using NA14 100y precipitation at storm center location
- SPAS 1184_1 – Clyde, TX 1981 - Revised GTF using NA14 100y precipitation at storm center location
- SPAS 1463_1 –Alvin, 1979 - Revised GTF using NA14 100y precipitation at storm center location
- SPAS 1464_1 – Houston, TX 2001 - Revised GTF using NA14 100y precipitation at storm center location
- SPAS 1582_1 – Broome, TX 1936 - Revised GTF using NA14 100y precipitation at storm center location
- SPAS 1591_1 – Hearne, TX 1899 - Revised GTF using NA14 100y precipitation at storm center location; Added the 1.10 tropical storm GTF cap.
- SPAS 1601_2 – Dinero, MX 1967 - Revised GTF using NA14 100y precipitation at storm center location
- SPAS 1667_1 – Harvey, 2017 - Revised GTF using NA14 100y precipitation at storm center location

## PMP Version 1.4a

Sensitivity version to refine transposition limits of several key events

**Local Storms:**

- SPAS 1226_1 – College Hill, OH 1963 – Applied a GTF cap of 1.20 in effort to better match surrounding depths
- SPAS 1295_3 – Hale, CO May 1935 – Applied a GTF cap of 1.00 in effort to better match surround depths over zone 11
- SPAS 1434_1 – Holt, MO 1947 – Extended the southern limit from 33°N to 32°N to provide a smoother transition
- SPAS 1495_1 – Cheyenne, OK 1934 – Applied a GTF cap of 1.00 in effort to better match surround depths over zone 9

**Local/Tropical Hybrid Storms:**

- SPAS 1592_1 – Thrall TX, 1921 – Applied NWS transposition limits (south of 34°N)

## PMP Version 1.4b

The following changes were applied to v1.4:

**Local Storms:**

- SPAS 1226_1 – College Hill, OH 1963 – Applied a GTF cap of 1.2 in effort to better match surrounding depths

- SPAS 1295_3 – Hale, CO May 1935 – Applied a GTF cap of 1.0 in effort to better match surround depths over zone 11; Added zone 12
- SPAS 1427_1 – Boyden, IA 1926 – Applied a GTF cap of 1.2 in effort to better match surrounding depths
- SPAS 1434_1 – Holt, MO 1947 – Include all of zone 9 and add zone 5 with a GTF cap of 1.2 to provide continuity over LA
- SPAS 1495_1 – Cheyenne, OK 1934 – Applied a GTF cap of 1.00 in effort to better match surround depths over zone 9

## PMP Version 1.5

The following changes were applied to v1.4b:

### Local Storms:

- SPAS 1495_1 – Cheyenne, OK 1934 – Applied a GTF cap of 1.00 in effort to better match surround depths over zone 9. Removed elevation constraint to allow transposition into all of zones 11 and 12 to provide spatial continuity over western OK.

### Tropical Storms:

- SPAS 1179_1 – Albany, TX 1978 – Added zone 5 to provide continuity to the coast for short-duration small area-sizes

# Appendix J
## Project Review Board Memos

Applied Weather Associates

PO Box 175
Monument, CO 80132
(719) 488-4311
http://www.appliedweatherassociates.com

April 11, 2019

Memo for Record

To: Oklahoma-Arkansas-Louisiana-Mississippi Review Board

Subject: Removal of Big Rapids, MI September 1986 storm, SPAS 1242

## Introduction

This memo provides background and reasoning for AWA's recommendation of removing the Big Rapids, MI September 1986 (SPAS 1242) storm event from the final short storm list and PMP development.

## Background

Applied Weather Associates has completed several sensitivities related to the use and removal of the Big Rapids, MI September 1986 (SPAS 1242) storm. These were based on discussions regarding this storm and its effects on PMP during review board meeting 2 and 3. During those discussions it was shown that the storm may not be transpositionable to any portion of this study. This was demonstrated by the excessive total adjustment factors (TAF) resulting from a high in-place maximization factor (IPMF) and excessively large Geographic Transposition Factor (GTF). The combinations of these two factors often resulted in TAF adjustments greater than 2.00. In other words, the storm that is controlling of PMP was being increased by 48% in place, then approximately 50% more by moving it from Michigan to Oklahoma and Arkansas. This resulted in likely inappropriate adjustments being applied to the storm.

Sensitivities were performed to develop a reasonable way to investigate this storm and determine whether it should be kept in the study or not. Sensitivities applied included capping the GTF factor at 1.40 and 1.20, using the GTF contours for transposition limits, and removal of the storm. Results of these sensitivities demonstrated that the storm should not be used in this study. The following support this recommendation:

- Use in Oklahoma and Arkansas would be beyond NWS transposition limits applied to similar storms in similar regions
- The synoptic meteorological storm environment was different than would occur over southern Great Plains in September
- Many other general storms in and nearby the region provide similar depths and greater certainty
- Adjustment factors significantly increasing the storm, more than doubling what was observed because of high IPMF and high GTF

- Large TAF result in the storm likely changing the storm significantly from what was observed and therefore violating the tenant of PMP that the storm dynamics are constant and unchanged after adjustments
- Resulting values still greater than HMR 51 at some area sizes/durations even after removal

Figure 1 displays a map of the General storm PMP for 5,000-square miles and 48-hour duration from the most recent version of PMP with the Big Rapids, MI September 1986 (SPAS 1242) storm included, while Figure 2 displays a map of the General storm PMP for 5,000-square miles and 48-hour duration with the Big Rapids, MI September 1986 (SPAS 1242) storm removed. This clearly shows that the spatial patterns are slightly improved and provide a more accurate fit without the storm included.

Figure 3 displays a map of the PMP compared to HMR 51 PMP for 10,000-square miles and 48-hour duration from the most recent version of PMP with the Big Rapids, MI September 1986 (SPAS 1242) storm included, while Figure 4 displays a map of the PMP compared to HMR 51 PMP for 10,000-square miles and 48-hour duration with the Big Rapids, MI September 1986 (SPAS 1242) storm removed. The region where the Big Rapids, MI September 1986 (SPAS 1242) storm controls (southern Kansas, northern Oklahoma, northern Arkansas) are significantly greater than HMR 51, by more than 30% in some cases. After removal of the storm, the difference from HMR 51 is reduced, but the values are still greater.

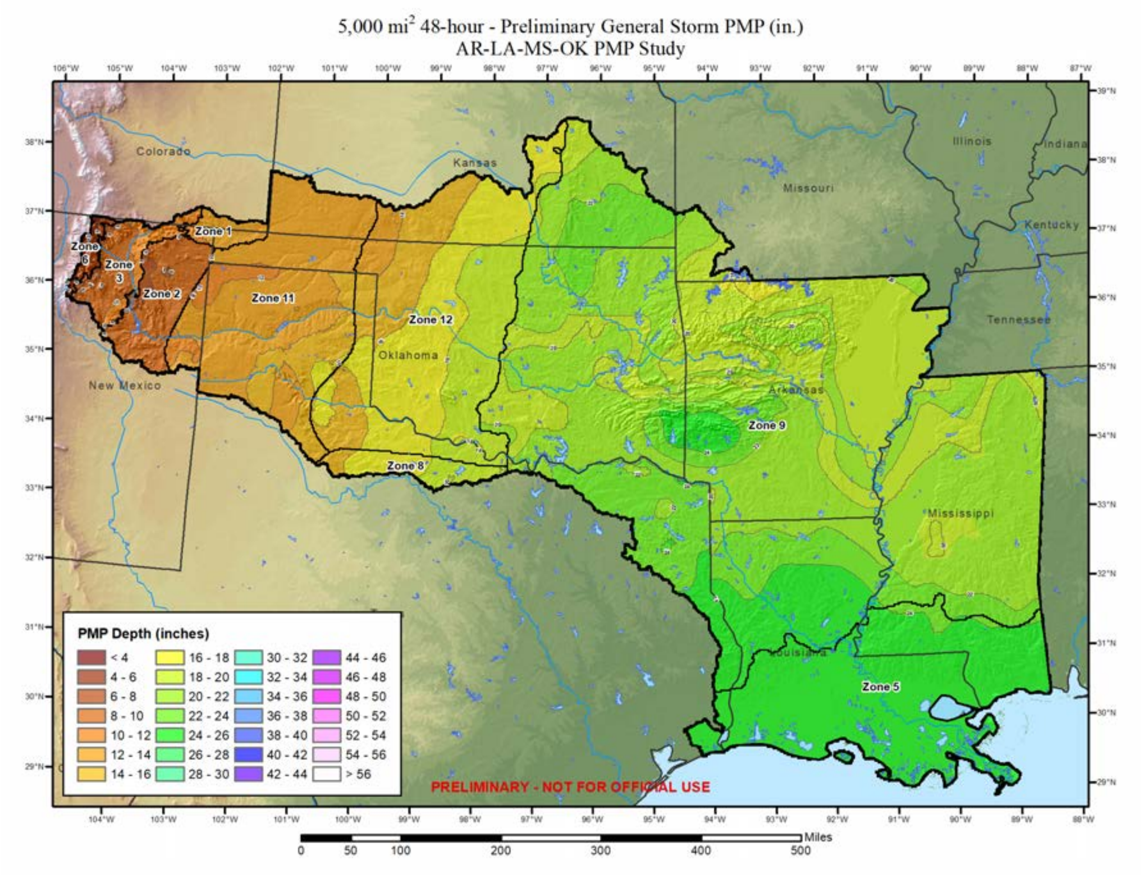


**Figure 1:** General storm 5000-square mile, 48-hour PMP with Big Rapids, MI included

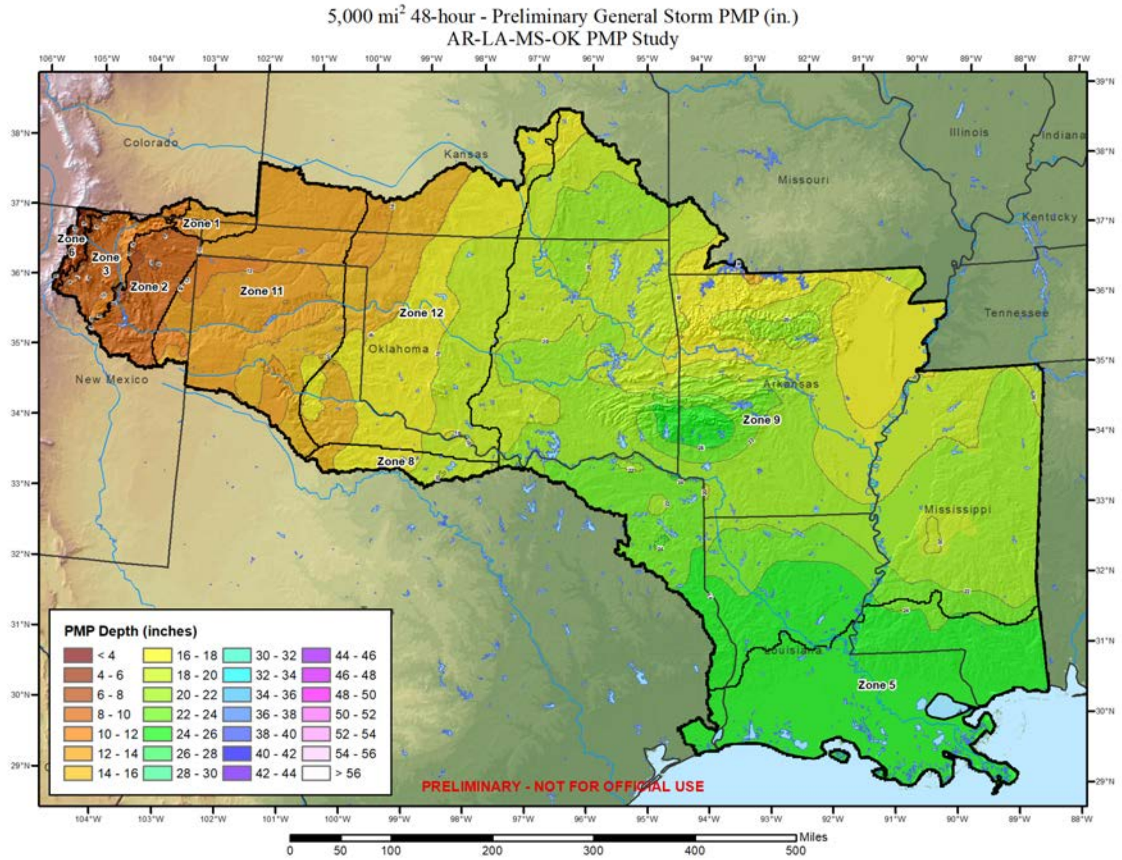


**Figure 2:** General storm 5000-square mile, 48-hour PMP without Big Rapids, MI

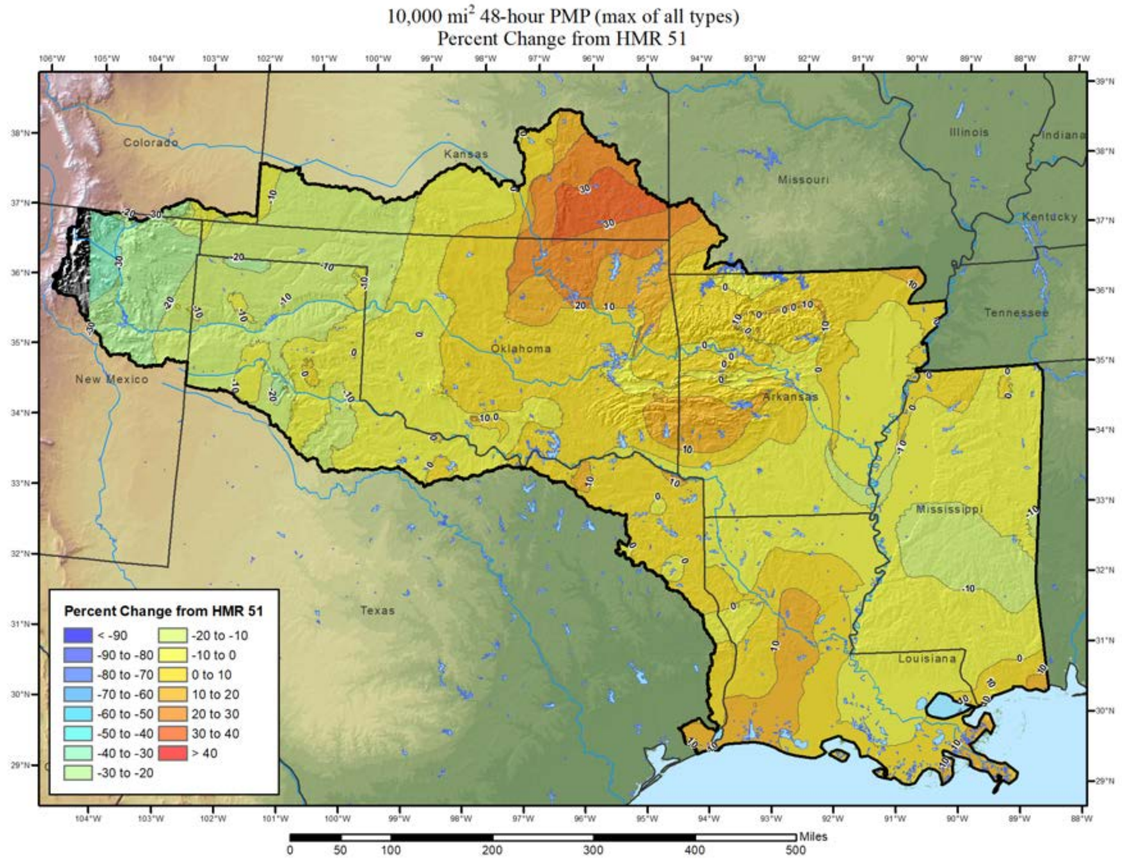


**Figure 3:** Comparison of PMP depths against HMR 51 with Big Rapids, MI

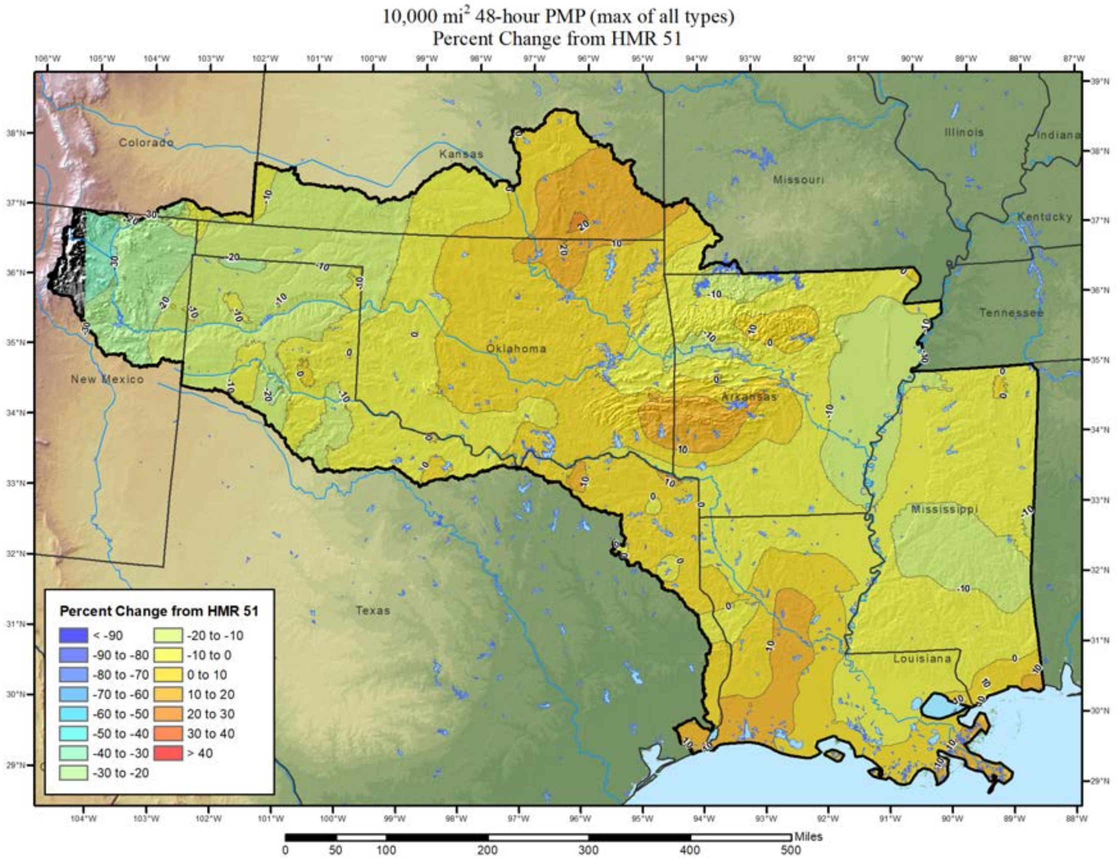


**Figure 4:** Comparison of PMP depths against HMR 51 without Big Rapids, MI

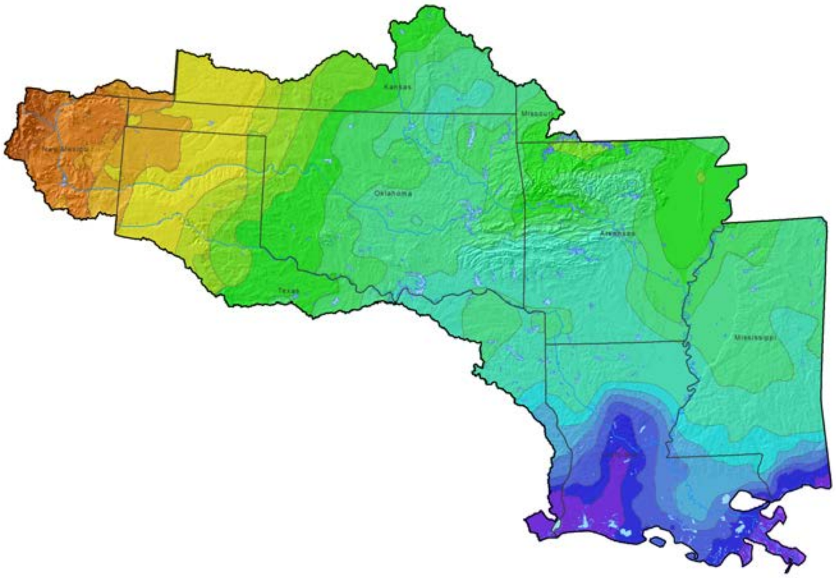
# Appendix K

## Project Review Board Letter

# Letter of Endorsement for

## *Regional Probable Maximum Precipitation Study for Oklahoma-Arkansas-Louisiana-Mississippi*

## By Applied Weather Associates LLC



Prepared for
Arkansas Dept. of Natural Resources
Louisiana Dept. of Transportation & Development
Mississippi Dept. of Environmental Quality
Oklahoma Water Resources Board

**August 2019**

## *Independent Review Team*
### *PMP Study for OK-AR-LA-MS*

**Background**

In September of 2017, a proposal selection meeting was convened by the Arkansas Department of Natural Resources in Little Rock, Arkansas to reach a consensus on a vendor to perform a Probable Maximum Precipitation (PMP) Study for Arkansas, Mississippi, and Louisiana. The intent of this study was to produce updated PMP estimates for the three-state region that would supersede estimates from Hydrometeorological Report No. 51 (HMR 51). Applied Weather Associates (AWA) LLC was selected at that meeting to conduct the study. Subsequent to this selection, Oklahoma was added to the regional study at the request of the Oklahoma Water Resources Board and AWA was tasked with the following:

- Perform a climatological and meteorological analysis of significant storms that influence PMP values across the four-state region. This is to include analysis of local storms, general storms, and tropical storms. In addition to the major storms included in the HMR 51 analysis, consider all additional major storm events that have occurred in subsequent years.
- Utilize updated tools and methods for estimating extreme precipitation depth, area, and duration relationships for the four-state region.
- Provide a final report to include PMP maps in GIS format for the four-state region for area sizes from 1 square mile to 20,000 square miles and for durations from 1-hour to 72-hours. Note, however, that the GIS-based program can extend out to durations of 96-hours and 120 hours, and for area sizes to 100,000 square miles. Also, a GIS-based program will be provided to allow users to extract the exact PMP values for any location in the four-state region.
- Coordinate an independent review team for the study to include expertise in PMP and dam safety across the region.

The Arkansas Department of Natural Resources engaged the services of several committee members to form an Independent Review Team (IRT). Members of this team include Whitney Montague - Arkansas State Climatologist; Barry Keim - Louisiana State Climatologist/Professor at LSU; Mike Brown – Mississippi State Climatologist/Professor at Mississippi State University; Edward Knight – Louisiana Dam Safety Program Manager, Louisiana Department of Transportation and Development; Zachary Hollandsworth – Oklahoma Water Resources Board; Yohanes Sugeng - Oklahoma Water Resources Board; Dusty Myers – Chief, Mississippi Dam Safety Division; Andrew Cummings – Mississippi Dam Safety Division, Mississippi Department of Environmental Quality; and Stephen Smedley – Arkansas Dam Safety Engineer, Arkansas Natural Resources Commission. In addition, representatives of the Federal Energy Regulatory Commission and the Natural Resources Conservation Service attended meetings and provided input to the study.

Four formal meetings and several conference calls were held to hear progress updates by AWA and to participate in discussions regarding process and methods for the study. The four meeting dates and locations were as follows:

- February 7-8, 2018 – Jackson, Mississippi
- July 17-19, 2018 – Baton Rouge, Louisiana

- January 22-23, 2019 – Baton Rouge, Louisiana
- June 4-5, 2019 – Little Rock, Arkansas

**Conclusions Regarding Study Analyses**

At these meetings and on several conference calls, the IRT was charged with reviewing and assessing each phase of AWA's four-state PMP study and for providing oversight, as necessary, to confirm the study methods were consistent with accepted PMP theories and procedures. The study used the same principles as the National Weather Service's HMR reports and World Meteorological Organization's PMP Manual for in-place maximizations, transposition, etc. However, in several instances, AWA utilized newer techniques and updated datasets. However, the methods reflect the most current practices used for defining PMP, including comprehensive storm analysis procedures. These new and improved procedures include use of the following:

- Storm Precipitation Analysis System (SPAS) software
- HYSPLIT model to assist in determining upwind back trajectories to determine appropriate source regions of storm moisture
- NOAA Atlas 14 for purposes of transposing storms from their source region to the study area (called the Geographic Transposition Factor), which accounts for elevation differences as well as differences in moisture availability
- Geographical Information Systems (GIS) for mapping,
- Updated maximum dew point and sea surface temperature climatologies for storm maximization
- Contemporary understanding of storm formation and storm climatology across the four-state region, which aided in determining transposition limits.

The IRT notes that the application of these contemporary procedures has been accepted in other AWA studies throughout the United States. Although this study produced deterministic values, the IRT realizes that there is some subjectivity associated with PMP development procedures, such as selection of storms used for PMP, determination of storm adjustment factors, and storm transposition limits. The IRT provided guidance on appropriate storm transposition limits, while considering both the meteorological and geographic limitations associated with each respective storm.

Use of the Moisture Transposition Factor (MTF) has been controversial in past PMP studies due to potential double counting of moisture in conjunction with the geographic transposition factor (GTF). The current study set the MTF = 1.0, which nullifies its use in this study. The IRT concurred with this decision.

This study provides PMP estimates for local storms (intense, short duration), general storms (extra-tropical cyclones), and tropical storms (including hurricanes). An example of the 1-hour/1 square mile local storm PMP map for the region is shown in Figure 1, while Figure 2 depicts the 72-hour/10 square mile tropical storm PMP for the region.

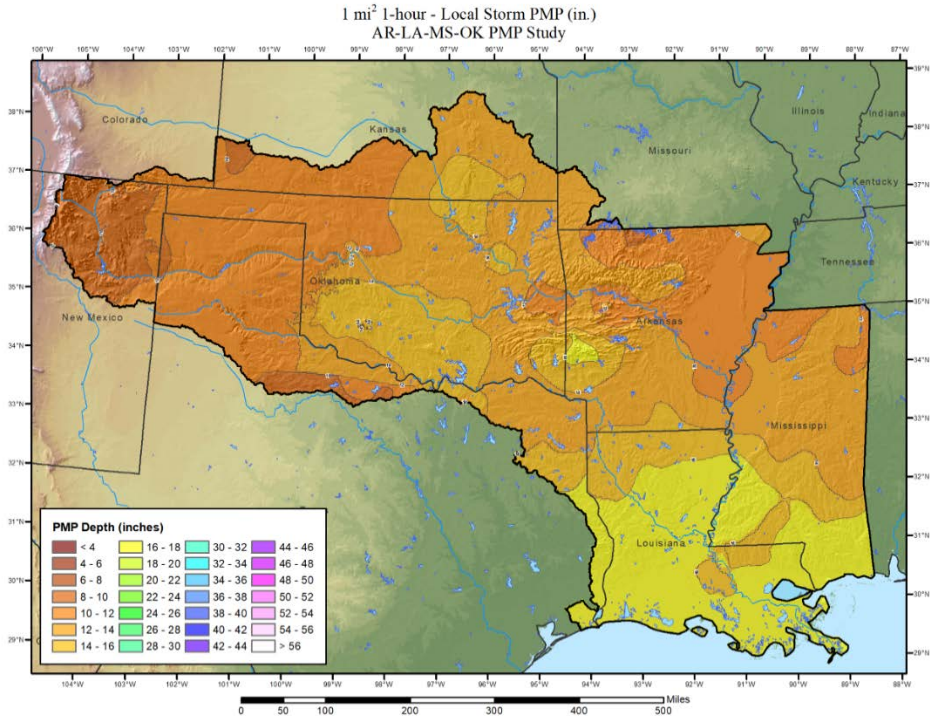


Figure 1. 1-hour/1 square mile local storm PMP map for Oklahoma, Arkansas, Louisiana, and Mississippi.

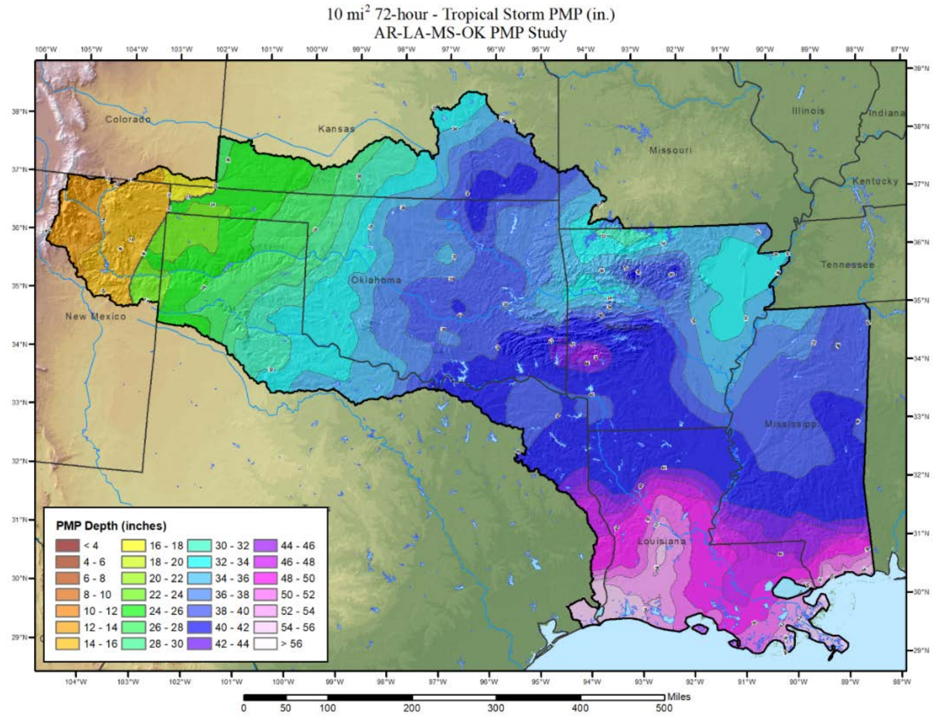


Figure 2. 72-hour/10 square mile tropical storm PMP map for Oklahoma, Arkansas, Louisiana, and Mississippi.

The IRT believes the PMP estimates of this study provide much improved results over those presently being utilized from HMR 51. The results include a longer period of record than that implemented in HMR 51, and the results have greater geographic resolution, thereby taking into account elevation differences across the region. As such, the IRT accepts AWA's estimates for PMP for use across the states of Oklahoma, Arkansas, Louisiana, and Mississippi. We also note that as future storms occur, the PMP will be updated by AWA.

The IRT performed the duties described above, but it should be noted that we acted in an advisory capacity only. Specifically, no calculations were performed by the IRT, nor were detailed reviews of calculations performed by the IRT. The IRT's expectation was that AWA utilized adequate quality assurance and control procedures to provide confidence that the calculations were performed accurately and without error. As such, results and conclusions in this report are based upon results from Applied Weather Associates, LLC. We, the oversight committee, used our best professional judgment in evaluating their work. We note that the final PMP estimates are based on storms from the historical record of the past century and more, with the underlying assumption that this record across the eastern United States yields insight into the PMP for the four-state region, including Oklahoma, Arkansas, Louisiana, and Mississippi. As such, we do not make any warranty, express or implied, regarding use of any information or method shown in the *Regional Probable Maximum Precipitation Study for Oklahoma-Arkansas-Louisiana-Mississippi* report, or assume any future liability regarding use of any information or method contained therein.

Respectfully submitted,

Barry Keim

Barry Keim

Edward Knight

Edward Knight

Zachary Hollandsworth

Yohanes Sugeng

Stephen C Smedley

Stephen C. Smedley, P.E.

Johnathon Atkins

Johnathon Atkins